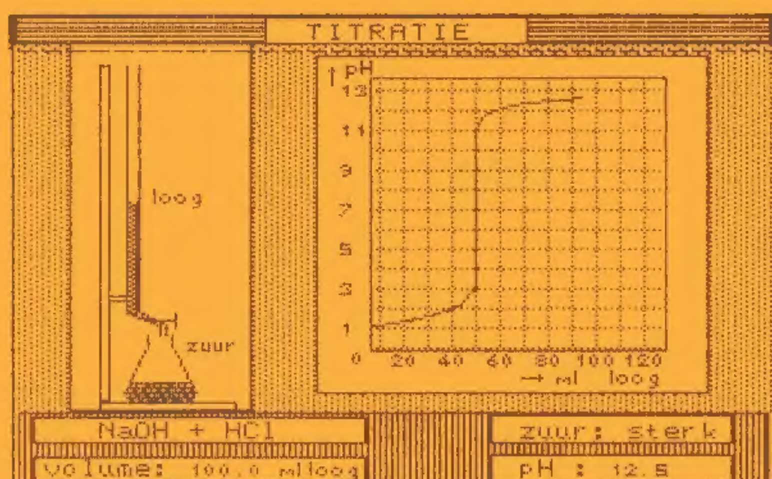
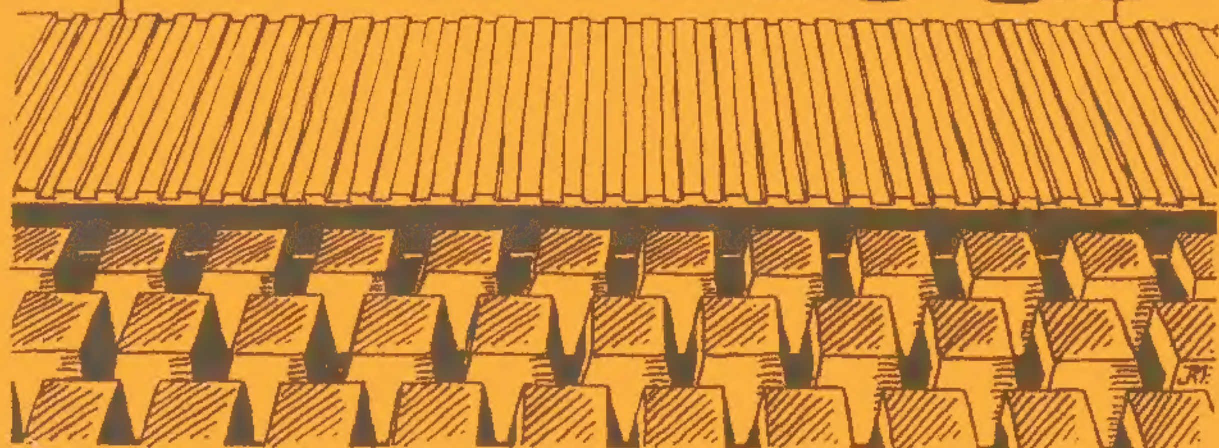
ATOM NIEUWS

JAARGANG

NUMMER





ATOM NIEUWS UIT DE FEDERATIE

Bestuur:

Voorzitter:	Secretaris:	Penningmeester:
N. Stad	N. Schoone	Th. van Kempen
Plataanweg 47	Sluispad 55-B	Het Puyven 71
1544 PB Zaandijk	1521 EZ Wormerveer	5672 RB Nuenen
Tel. 075-280808	Tel. 075-286624	Tel. 040-836210

Clubwinkel:	Hard-ware cie.:	Redaktie Atom Nieuws:
N. Stad	P. Ehrlich	H. de Ruiter
Plataanweg 47	Roostenlaan 266	Polarisstraat 25
1544 PB Zaandijk	5644 BS Eindhoven	8303 AC Emmeloord
Tel. 075 - 280808	Tel. 040 - 114183	

Contributie 1985: fl. 60,00
Leden Buitenland: fl. 75,00

Giro 5244293 Bank 52.84.69.010
Beide t.n.v. Penningmeester Atom Computerclub Nuenen

Redaktie Atom Nieuws:	Redaktieadres:
Joop Ballijns	Postbus 1373
Harry de Ruiter	8001 BJ Zwolle
Ed Schijf	
Evert van Schothorst	
Jaap van de Veen	
Gerhard Visser	
Wibo Visser	

Ledenadministratie:	Datasheets:	Drukwerkarchief:
S. van Leeuwen	G. Akkermans	F. Monsanto
Kompasstraat 32	Wikke 1	C. Kohlerstraat 97
1973 PX IJsselmuiden	1273 BR Huizen	7558 VC Hengelo
Tel. 02550 - 22435	Tel. 02152 - 60294	

Uiterste datum inlevering copy:
nr. 7/8 8 - 11 - 1985

DE CLUB-WINKEL:

Geheugenkaart; 16 kByte extra in de ATOM	fl. 40,00
Schakelkaart; meerdere EPROM's op Axxx (incl. 74LS133)	fl. 50,00
Programmerkaart; zelf programmeren van EPROM's	fl. 21,00
Minischakelkaart	fl. 16,00
Herdruk ACORN NIEUWS 1982; 97 pag. wetenswaardigheden	fl. 6,00
Jaargang 1983; totaal ruim 450 pag.	fl. 30,00
Jaargang 1984; totaal ruim 650 pag.	fl. 35,00
ATOM-WARE deel 1; machinetaal op de ATOM, 98 pag.	fl. 6,00
ATOM-WARE deel 2; het diskdrive operatingsystem, 68 pag.	fl. 5,00
ATOM-WARE deel 3; het monitor operating system, 80 pag.	fl. 5,00

Levering:
Via uw regionale penningmeester of rechtstreeks bij Atom Computerclub te Nuenen, uitsluitend na voorafgaande betaling (vermeerderd met fl. 4,00 voor portokosten).
Indien u 14 dagen na afschrijving van uw bank- of girorekening nog niets hebt ontvangen, informeer dan bij de clubwinkel.

ATOM NIEUWS is een uitgave van de fed. atom computerclubs n/b en verschijnt 6 - 8 keer per jaar.

Van de redaktie

Zo, het eerste officiele woord van de nieuwe redaktie staat geschreven. Voor de redaktie betekent dit dat het jachtseizoen geopend is. Op U wel te verstaan en vooral op diegenen die een ATOM hebben en er iets mee (willen) doen: gebruikers dus.
U kent zeker het spreekwoord "wat de boer niet kent dat (vr)eet hij niet".

Even terzijde om heftige discussies te voorkomen.
Tegenwoordig schijnen boeren iets samen te doen met computer-fabrikanten (die boeren wel of niet goed) en dus bestaat de mogelijkheid veel lof te oogsten.
Ten tweede het woord eet cq. vreet. Wij geven de voorkeur aan het woord eet omdat het woord vreet impliceert dat de moderne boer zich beweegt op het niveau van het ouderwetse varken. Nou, wij weten wel beter! By the way, hoe zou die boer anders het voer in de trog van het varken krijgen? (nivo, he).

Om nu de draad van het oorspronkelijke verhaal weer op te pakken de vraag of u wel eens met een bovengenoemd spreekwoordelijk figuur bent wezen chinezen?
Ja zegt u, en hij bestelde een portie nasi.
Dat zal lekker gesmaakt hebben. Wat zegt u, niet???
U zegt dat alles wat het heerschap niet bekend voorkwam rond z'n bord gedeponeerd werd, zodat het al gauw leek alsof hij van de tafel zat te eten in plaats van z'n bord.
Nou uh, uh .... smakelijke voortzetting dan.
We weten allemaal dat deze maaltijd snel beeindigd zal zijn.
Bovenstaande heeft, zoals u begrepen hebt, niet de bedoeling boeren en chinezen in een kwaad daglicht te stellen. Integendeel.
De bedoeling is u te laten inzien dat eenvoud van belang is maar dat ook ingewikkelder zaken hun plaats moeten hebben. Zo komt eenieder aan z'n trekken.
De taak van de redaktie is het een en ander op de juiste wijze samen te voegen en de lawine van artikelen die we verwachten op de juiste wijze te rangschikken.
Dus, hebt u (nog) iets liggen, een leuk programma, iedee of ontwerp, stuur het ons toe!!!!!!!!!!!
Ook voor klachten, kopstoten, rellen schoppen, wurgingen en schouderklopjes kunt u bij ons terecht. Stuur uw bijdrage naar:
Redaktie Atom Nieuws
Postbus 1373
8001 BJ Zwolle

Met vriendelijke groet,

het re(d)aktie-team.

P.S.
Onze Don Atom van de redaktie vroeg zich af of er ook vrouwen binnen de federatie actief zijn. Zo ja, laat eens wat eh, eh van je zien.

VAN DE VOORZITTER

Met ingang van dit nummer van ATOM NIEUWS hebben we een nieuwe redaktie.
Henk Reinders zag zich gedwongen om te stoppen met het vervaardigen van ons clubblad.
Via deze weg wil ik hem bedanken voor al het werk dat hij de afgelopen twee jaar voor ons clubblad heeft gedaan, door hem is het uiterlijk tot een hoog peil gegroeid.

HENK BEDANKT VOOR JE WERK.

Tevens wil ik de nieuwe redaktie veel sterkte en plezier in het werk toewensen. Maar zonder hulp van de leden en de regio's komt een clubblad niet vol, dus een ieder die copy heeft roep ik op om dit of via hun regio of direkt aan de redaktie te zenden.
Copy kan op de volgende manieren bij de redaktie komen :

1) per schijf (40 tracks 10 sectoren enkelzijdig)

2) per cassette (het liefst 300 baud)

3) op papier (A4-formaat, tekst afmeting 17 * 25 cm, en goed zwart een nieuw lint gebruiken.)

Tekstfiles het liefst met de WORD-PACK gemaakt, of een basic programma met direkt achter de regelnummers de tekst.
Maar absoluut geen formatgegevens in de tekstfile, want die zijn voor elke printer anders.

Als een ieder zich aan deze drie regels houdt, dan is het voor de redaktie eenvoudig om copy te verwerken, en ik vind ook dat als er niet aan deze drie regels wordt voldaan, dat de redaktie het recht heeft om copy aan de kant te leggen.
De afmetingen 17 * 25 cm komt overeen met 59 regels van 65 tekens.
Laten we ons hier allemaal aan houden.
Dit stukje is volgens regel drie ingeleverd en dat levert in het geheel geen problemen op.

Dan zou ik als laatste willen zeggen :
Laten we het gezellig houden, want we zijn tenslotte een hobbyclub.

Nico Stad

BEEINDIGEN VAN HET LIDMAATSCHAP.

Van de regionale statuten luidt artikel 6.3 1ste alinea:
Opzegging van het lidmaatschap door het lid of door de vereniging kan slechts geschieden tegen het einde van een verenigingsjaar en met inachtneming van een opzegtermijn van vier weken.

Indien u geen lid meer wenst te blijven, verzoek ik u dit SCHRIFTELIJK te melden aan:
Atom Computerclub, Het Puyven 71, 5672 RB Nuenen.

CONTRIBUTIEBETALING 1986.

Ingesloten treft u de acceptgiro aan voor de contributiebetaling 1986.
Hierop zijn uw regio, lidmaatschapsnummer, naam en adres voorbedrukt.
Indien dit niet juist is verzoek ik u veranderingen DUIDELIJK EN IN BLOKLETTERS op de achterzijde te vermelden.

De contributie moet voor 15 december a.s. op de girorekening van de federatie bijgeschreven zijn. Dit in verband met tijdige verzending van het eerste nummer van Atom Nieuws van 1986 door de drukker.

Indien u na 10 december betaalt, dan kunt u GEEN GEBRUIK MEER MAKEN VAN DE ACCEPT-GIRO maar moet u fl. 65,= naar de giro- of bankrekening van de federatie overmaken, om het (de) reeds verschenen nummer(s) nagezonden te krijgen.

WACHT DUS NIET TE LANG EN MAAK DE CONTRIBUTIE METEEN OVER.

De penningmeester.

UITLEG BIJ INHOUD SCHIJVEN PROGRAMMA.

Dit progamma heb ik gesitueerd in het battery-backup systeem van de geheugenkaart tussen #4000 en #6800. Ik heb hier ruimte voor 17 schijven. Als je meer schijven hebt,, dan kan je natuurlijk ook op een andere geheugenplaats starten. Je moet dan wel het programma enigszins aaanpassen. Wanneer je op #2800 start dan is er plaats voor 29 schijven. Je moet dan in regel 140 veranderen P=#2800 en in regel 560 wordt #4000 verandert in #2800.

Als men het programma voor het eerst start dan moet men eerst ?U op 0 zetten. Dit doet men door voor het runnen eerst in te tikken ?#45FF=0 (of ?#2dFF=0).
Na het runnen krijgt men het menu te zien. Om de schijven in het bestand te krijgen moet men nu stuk voor stuk (op volgnummer) de schijven in de drive doen en dan op de T(oevoegen) drukken, waarna het programma vraagt "NIEUWE SCHIJF J/N" bij "J" komt de inhoud van de schijf op het scherm en in het geheugen. Een volgende schijf weer beginnen met "T" etc.
Men kan op drie manieren de inhoud van de schijven bekijken.1e met N(ext), dan wordt de volgende schijf bekeken. 2e met P(revius), dan wordt de inhoud van de vorige schijf teruggehaald. 3e tenslotte, rechtstreeks het nummer van de schijf die men wil zien, intoetsen, altijd met twee cijfers, dus b.v. 07,24 of 01 (zonder return).
Is de inhoud van een schijf verandert en wil men deze schijf aanpassen dan drukt men op de V(erbeteren), als de betreffende schijf-inhoud op het scherm staat. Het programma vraagt dan weer "NIEUWE SCHIJF J/N" en werkt verder identiek aan T(oevoegen).

Heeft u alles gezien en eventueel het een en ander gewijzigd en u wil uit dit programma druk dan op E(inde). U moet dan wel zorgen dat de schijf waar dit inhoud-programma op staat, in de drive zit. Dit is nodig omdat, als er schijven gewijzigd of toegevoegd zijn, het programma zich zelf saved met de aangepaste schijven-inhoud, waarbij het oude bestand overschreven wordt.

Met viendelijke groeten,

Bram Thijsse
Albert Verweijstraat 20
2406 VX Alphen a/d Rijn

=======================

=================

===========

=====

ASTROLOGIE
==========

Zoals U reeds begrepen zult hebben gaat dit stukje over astrologie en het maken van horoscopen. De astrologie mag zich verheugen op een eeuwenlange belangstelling. Reeds in de tijd van de farao's kwamen in Egypte astrologen voor. Deze pretenteerden aan de hand van de stand van de planeten gunstige of ongunstige tijdingen te kunnen voorspellen. Deze astrologen hadden niet de beschikking over de moderne hulpmiddelen zoals wij deze kennen. Toch wisten ze vrij nauwkeurig de stand van de planeten te berekenen. Door de eeuwen heen is er altijd belangstelling voor astrologie geweest, maar de laatste jaren is de belangstelling voor astrologie sterk toegenomen. De meesten onder U zullen zich al dan niet serieus met astrologie hebben bezig gehouden. Iedereen zal wel eens zijn horoscoop lezen zoals deze vaak in dag- en weekbladen wordt gepubliceerd. Diegenen die al eens aan astrologie en het maken van horoscopen hebben gedaan, weten wat een ontstellende hoeveelheid rekenwerk er aan te pas komt. Het navolgende programma is een horoscoopberekeningsprogramma. Het neemt al het vervelende rekenwerk uit handen en maakt tot slot ook nog eens de horoscooptekening.
Het programma is voor een gedeelte een vertaling van een horoscoopprogramma, welke het computertijdschrift "Computerplus" in drie afleveringen publiceerde. Het programma was oorspronkelijk bedoeld voor de Commodore 64. Bepaalde onderdelen van het programma moesten dan ook op geheel andere wijze opnieuw geprogrammeerd worden.

Over tijdstippen in een horoscoop valt heel wat te zeggen. Alles staat of valt met de invoer van de juiste tijdstippen. Er zijn dan ook een aantal tijdsoorten. Een waarnemer stelt vast dat de zon op zijn hoogste punt staat. Het is dus ter plaatse 12 uur. Op dezelfde plaats 24 uur later is het dat weer. Dit is ware ZONNETIJD: in 24 uur wordt een volledige cirkel doorlopen (=15 graden per uur). De ware zonnetijd verschilt dus van plaats tot plaats. Daarom zijn er: TIJDZONES. De nulmeridiaan (die loopt door Greenwich) geeft de wereldtijd aan. Een gebied van 7 minuten en 30 seconden (7m30s) aan beide kanten van de meridiaan heeft dezelfde tijd.
Van 7m30s tot 22m30s Oosterlengte is het 1 uur later dan in de Greenwichzone (bv. Nederland). Maar meridianen en landsgrenzen vallen niet samen. Dat is ongemakkelijk. Daarom bestaan er STANDAARDTIJDEN. Heel West-Europa heeft de middelbare Europese tijd(=Greenwichtijd plus 1 uur).
Het geboortetijdstip zoals dat geregistreerd wordt door de burgelijke stand is in MIDDELBARE PLAATSELIJKE tijd (iemand "klokt" bij de bevalling op een horloge dat de tijd aangeeft die in die tijdzone van toepassing is). Meestal is dat dus middelbare Europese tijd.

Middelbare plaatselijke tijd is gebaseerd op een gemiddelde; dit gemiddelde heeft twee correcties nodig in verband met afwijkingen die ontstaan door draaiingen van zon en aarde. Daarom voeren we het begrip STERRETIJD in.
De sterretijd wordt ten opzichte van een vast hemelobject (nul graden Ram = lentepunt) gemeten. Om het geboortetijdstip in sterretijd uit te drukken doen we het volgende: eerst bepalen we

de sterretijd op het moment dat het nul uur middelbare Greenwichtijd is op de geboortedag. Vervolgens vermenigvuldigen we de middelbare plaatselijke tijd van het tijdstip van de geboorte (decimaal genoteerd) met het getal 1.00273791 (dit is de correctie voor de bewegingen van aarde en zon). Tel de uitkomsten bij elkaar op. Dit levert het geboortetijdstip in sterretijd uitgedrukt. deze sterretijd hebben we later nodig bij het berekenen van de huizen. De Greenwichtijd is nodig voor de berekening van de stand van de planeten.
Van belang voor het berekenen van horoscopen zijn de invoergegevens. In dit programma moeten achtereenvolgens ingevoerd worden:
NAAM - van de betreffende persoon
GEBOORTEDATUM - moet worden ingevoerd in cijfers. Het jaartal moet voluit. Tussen de cijfers moeten punten.
GEBOORTETIJDSTIP- is de werkelijke tijd van de geboorte (Kan tegen betaling worden opgevraagd bij de Burgelijke Stand)
ZOMERTIJD J/N - hier moet met J(a) of N(ee) op worden geantwoord.
TIJDZONE - hier moet een getal worden ingevuld gelegen tussen -12 en +12. Voor Nederland geld +1. In het algemeen geldt dat plaatsen met een oosterlengte "+" ervoor krijgen en plaatsen met een westerlengte een "-" ervoor.
GEBOORTEPLAATS - hier moet de plaats ingevuld worden. Van de plaatsen in onderstaande tabel zijn de lengte- en breedtegraden al bekend in het programma, dat betekent dat men alleen de plaats hoeft in te vullen en geen antwoord meer hoeft te geven op welke lengte- en breedtegraad de plaats is gelegen. Als U een plaatsnaam toe wilt voegen, moet U dit doen in de dataregels aan het eind van het programma, voor de laatste dataregel. Achter de plaatsnaam moeten dan achtereenvolgens de lengtegraden en de breedtegraden worden gezet.
LENGTEGRAAD - hier moet de lengtegraad in de vorm GG.MM (Graden.Minuten) worden ingevuld (tussen 0 en 360 graden).
BREEDTEGRAAD - hier moet de breedtegraad in de vorm GG.MM (Graden.Minuten) worden ingevuld (zuiderbreedte moet negatief worden ingevuld!).

Hieronder volgen enkele plaatsen in Nederland, met daarachter de breedtegraad en lengtegraad.

	Noorder-breedte	Ooster lengte
Alkmaar	52.37'	04.45'
Amsterdam	52.21'	04.55'
Assen	52.29'	06.34'
Breda	51.34'	04.48'
Eindhoven	51.26'	05.30'
Enschede	52.12'	06.53'
Den Haag	52.05'	04.18'
Groningen	53.13'	06.33'
Haarlem	52.22'	04.38'
Den Bosch	51.40'	05.20'
Hoorn	52.38'	05.04'
Leeuwarden	53.12'	05.47'
Maastricht	50.51'	05.41'
Middelburg	51.30'	03.37'
Roosendaal	51.32'	04.28'
Rotterdam	51.55'	04.30'
Tilburg	51.33'	05.07'
Utrecht	52.05'	05.08'
Wageningen	51.58'	05.40'
Zwolle	52.30'	06.05'

Aan de hand van de lengte- en breedtegraden van deze plaatsen kunt U zelf ongeveer de coordinaten van uw eigen geboorteplaats bepalen.
We zullen de werking van het programma aan de hand van een voorbeeld verduidelijken.
VOORBEELD: Iemand is geboren op 27 september 1961 te 05.57 uur. De geboorteplaats is Amsterdam. Zoals U net hebt kunnen lezen is de breedtegraad voor Amsterdam 52.21' en de lengtegraad 04.55'. Op de vraag "TIJDZONE" vult U +1 in. Op de vraag "PRINTER J/N" dient U met J of N te antwoorden. Bij J dient de printer wel aan te staan, anders loopt het programma vast. Vervolgens worden de ingevoerde gegevens geprint alsmede de sterretijd. Daarna duurt het ongeveer 20 seconden voordat de Ascendant en MC geprint worden. Dat ziet er als volgt uit:

```
ASCENDANT: 26:27  MAAGD
MC       : 25:20  TWEELINGEN
```

Hierna volgen de huizen. U zult zien dat HUIS 1 gelijk is aan de ascendant en dat huis 10 gelijk is aan de MC. Van de huizen worden alleen de huizen 10, 11, 12, 1, 2 en 3 berekend. De andere zes huizen zijn tegenover liggende huizen (in een cirkel gezien) zodat deze 180 graden of zes tekens (elk teken is 30 graden) van elkaar verwijderd liggen. Er zijn ongeveer 20 manieren om huizen te berekenen. Wij hebben gekozen voor de puur wiskundige manier van berekenen en dat is die van Placidus. Deze heeft alleen als nadeel dat er dicht bij de beide polen afwijkingen kunnen ontstaan. Voor het gegeven voorbeeld zien de huizen er als volgt uit:

H U I Z E N (PLACIDUS)

HUIS 10	:	25:20	TWEELINGEN
HUIS 11	:	01:54	LEEUW
HUIS 12	:	02:20	MAAGD
HUIS 1	:	26:27	MAAGD
HUIS 2	:	19:42	WEEGSCHAAL
HUIS 3	:	19:18	SCHORPIOEN
HUIS 4	:	25:20	BOOGSCHUTTER
HUIS 5	:	01:54	WATERMAN
HUIS 6	:	02:20	VISSEN
HUIS 7	:	26:27	VISSEN
HUIS 8	:	19:42	RAM
HUIS 9	:	19:18	STIER

Hierna wordt de stand van de planeten berekend. De Zon en de Maan worden gemakshalve tot de planeten gerekend. De standen voor het gegeven voorbeeld zijn:

PLANEET		POSITIE	
ZON	:	03:51	WEEGSCHAAL
MERCURIUS	:	29:42	WEEGSCHAAL
VENUS	:	04:20	MAAGD
MARS	:	26:53	WEEGSCHAAL
JUPITER	:	27:18	STEENBOK
SATURNUS	:	23:19R	STEENBOK
URANUS	:	28:26	LEEUW
NEPTUNUS	:	09:41	SCHORPIOEN
PLUTO	:	08:43	MAAGD
KNOOP	:	25:05	LEEUW
MAAN	:	10:09	STIER

Met KNOOP wordt hier de klimmende knoop van de maanbaan bedoeld. Indien een planeet Retrograad of Directgaand is wordt dat aangegeven met een "R" of een "D" achter de cijfercombinatie. Vervolgens worden de ASPECTEN van de planeten berekend. De aspecten zijn de hoeken tussen 2 planeten in de dierenriem. Een hoek van 0 graden (de planeten liggen dus op een lijn ten opzichte van de waarnemer) heet een conjuntie. Hoeken van 60, 90, 120 en 180 graden heten respectievelijk sextielen, kwadraten (of vierkanten), driehoeken en opposities. Een verdere onderverdeling is hier ook nog mogelijk. Zo heet een hoek van 30 graden een halfsextiel, een hoek van 45 graden een halfkwadraat en een van 135 graden een anderhalfkwadraat. Al deze hoeken worden met een zekere nauwkeurigheid bepaald en geplaatst in een dierenriemtekening. Een bepaalde marge (speling) die Orb genaamd wordt is hierbij toegestaan. Naarmate de orb kleiner is zijn ook het aantal interpretatiemogelijkheden van de horoscoop kleiner. Door de desbetreffende DATA regels in het programma te wijzigen kan men ook de Orb veranderen. Samengevat zijn de volgende aspecten mogelijk:

CONjunctie	0 graden
HalfKWadraat	45 graden
SEXtiel	60 graden
72	72 graden
KWAdraat	90 graden
DRIehoek	120 graden
AnderhalfKWa	135 graden
OPPositie	180 graden

Als U de printer heeft aanstaan dan krijgt U een zogenaamde aspectentrap. Als U deze niet heeft of niet heeft ingeschakeld verschijnt er op het beeldscherm elk aspect dat er gemaakt wordt. De in het programma gegeven printroutine voor het maken van de aspectentrap is bedoeld voor de Seikosha GP 250X. Als U zo'n aspectentrap op uw printer wilt maken dan moet U de desbetreffende regels aanpassen.
Probeert U ook eens de volgende voorbeelden:
De geboorte van alledrie de voorbeelden is 1 juli 1979.

1. geboorte te Amsterdam, kloktijd 14.05 uur
   zomertijd: ja, verschil van een uur
   positie: 52.21' N.B.; 04.23' O.L.
   tijdzone: Middeleuropese Tijd, dat is 1 uur verschil met Greenwich (namelijk de 15' O.L. meridiaan) dus +1

2. geboorte te New York, kloktijd 08.05 uur
   zomertijd: ja, verschil van een uur
   positie: 40.45' N.B.; 73.57' W.L.
   tijdzone: E.S.T., dat is 5 uur verschil met Greenwich (namelijk de 75' W.L. meridiaan) dus -5

3. geboorte te Djakarta, kloktijd 19.05 uur
   zomertijd: nee
   positie 06.10' Z.B. (=-06.10); 106.49' O.L. (Java)
   tijdzone: sinds 1-1-1964 Standaard tijd van 105' O.L.-meridiaan. Verschil met Greenwich 7 uur. Dus +7 uur.

U zult zien, dat de posities van de huizen tussen de voorbeelden aanmerkelijk zullen verschillen, maar dat de uitkomsten van de planetstanden van alle voorbeelden hetzelfde zullen zijn. Dit komt omdat de planeetstanden naar Greenwichtijd worden berekend. Alleen de maan zal bij devoorbeelden steeds een andere plaats hebben, omdat die berekend wordt aan de hand van de plaats op aarde. Dit omdat de maan heel dicht bij de aarde staat, zodat je vanuit een andere plaats de maan in een andere richting ziet (natuurlijk zijn die verschillen heel erg klein). En in deze voorbeelden was de Greenwichtijd steeds dezelfde, terwijl de kloktijd aanzienlijk verschilden. De Greenwichtijd voor alle voorbeelden was 12.05 uur.

Tot slot van het programma wordt de horoscooptekening gemaakt. Dit gebeurt aan de hand van de berekende gegevens.
Wie kunnen dit programma gebruiken. Wij zouden zeggen "IEDEREEN". Er is echter nogal wat geheugenruimte voor nodig. Het programma zelf loopt van #2900 tot #55A3 en inclusief dimensionering tot #5938). U heeft 2K arrays nodig. Deze lopen in het programma van #9800 to #A000. Noodzakelijk is eveneens de P-CHARME. Tevens moet GAGS kunnen worden gebruikt. (voor de tekening). Er moet namelijk 2K worden gereserveerd voor "base" deze staat momenteel op #2000. Er wordt aan gewerkt om al de extra statements die nodig zijn voor dit programma in een box te stoppen en deze achter het programma te plaatsen. De statements worden dan herkend door de P-CHARME. Deze blijft dan noodzakelijk.
Het programma zal naar het bandjesarchief gaan. Voor DISK-gebruikers zal er waarschijnlijk een apart programma komen waarop meer graphics voorkomen.

Als U dit hele verhaal gelezen hebt zult U denken wat moet ik met zo'n berg gegevens. Deze gegevens zijn slechts de basis voor het belangrijkste en tevens ook het moeilijkste in de astrologie: het duiden. Duiden houdt in dat er aan de gegevens een betekenis wordt gegeven. De gegevens die U nodig hebt voor dit duiden kunt U vinden in boeken die in bijna elke bibliotheek te vinden zijn. Het is beslist niet noodzakelijk dat U in astrologie gelooft, maar probeert U het eens. U zult zien dat de resultaten in de meeste gevallen verbluffend zijn. Wij begrijpen dat dit verhaal niet helemaal volledig is, maar over dit onderwerp is zoveel te vertellen. Mocht U vragen hebben dan kunt U die gerust aan de auteurs stellen. Veel plezierige uurtjes toegewenst met dit programma.

Erwin van den Haak en Marcel van der Looy, Regio Noord-Holland

Een programma voor ATOMISTEN die van water houden (deuterium ??)

```
 10 PROGRAM BLADVULLING
 20 REM (C) HAMBURGERSOFT
 30 FUNCTION FNA(X,Y),A
 40 %A=SQR(X*X+Y*Y)
 50 A=%A
 60 %Z=(SIN(RAD(%A)*2)*200-%A)/5
 70 FNA=%Z
 80 FEND
 90
120 CLEAR 4
130 %X=128;%Y=96
140 FOR X=-640 TO 0 STEP 5
150 M=-512
160 FOR Y=-1212 TO 912 STEP 20
170 V=(Y+FNA(X,Y))/2
180 IF M<V THEN M=V;GOS.a
190 NEXT Y
200 NEXT X
210 END
220aPLOT13,((X/5)+%X),((V/5)+%Y)
230 PLOT13,((-X/5)+%X),((V/5)+%Y)
240 R.
```

Op de vergadering van 14 sept. j.l. is weer eens het begrip minimale Atom ter sprake gekomen en het wordt zo zoetjesaan tijd dat er van federatieve zijde bepaald wordt hoe een Atom er minimaal uit dient te zien om op een prettige manier mee te doen met het landelijke gebeuren.
Het is nl. zo dat je niet tot in lengte van dagen kunt blijven terugvallen op de allerminimumste Atom met 1/2 k programma-ruimte. Aan de andere kant willen we ook niemand opjagen om zijn of haar Atom met alles wat ooit een keer is gepubliceerd uit te rusten. Ergens ligt natuurlijk een grens en die hebben we voorlopig op een naar onze mening zeer acceptabel en goed bereikbare plaats gelegd.
Voor de goede orde:
WE KUNNEN EN WILLEN NIEMAND OP WELKE MANIER DAN OOK VERPLICHTEN TOT BEPAALDE HANDELINGEN EN/OF MODIFICATIES.
We willen alleen een zgn. federatieve Atom definiëren waarmee de uitwisseling van programma's wat soepeler zal verlopen. Je hebt dan tenminste een redelijke basis waarop je kunt terugvallen. Elk programma dat op een federatieve Atom draait, behoeft wat dat betreft dan ook geen verdere uitleg. Worden er specifieke uitbreidingen verlangd (b.v. extra veel RAM of RAM op bijzondere locaties of speciale toolkits) dan kan dat in de beschrijving kort vermeld worden.

Alors, de federatieve Atom bestaat uit de door Acorn geleverde minimumconfiguratie plus:

a1. RAM van #2800 totaan #3BFF.

a2. graphics RAM van #8000 totaan #97FF.

a3. een floating point (EP)ROM al of niet met driewegwissel.

a4. een VIA (IC 1, 6522) met LINK 2 doorverbonden.

a5. een 1200 baud routine.

- -

ad.1 het RAM gebied #2800-#3BFF kan eenvoudig in de voetjes in de Atom geprikt worden. Het RAM gebied #3C00-#3FFF wordt eventueel gestapeld zoals beschreven in AcornNieuws '82 nr.3.

ad.2 eenvoudig inprikken in Atom voetjes.

ad.3 het driewegwissel is beschreven in AcornNieuws '83 nrs.1&2. Het wissel wordt niet "verplicht" gesteld, omdat velen een echte ROM hebben die niet te veranderen is.

ad.4 eenvoudig inprikken en draadje leggen.

ad.5 vrijwel alle 1200 baud routines zijn uitwisselbaar. Ze zijn te vinden in diverse toolkits (Josbox, Program Power Toolkit, P-charme, Softtool, Screenrom), maar zijn ook beschikbaar als losse routines (Fastcos en Allcos (AcornNieuws '83 nr.1 p.59)).

Verder worden de volgende zaken zeer sterk aanbevolen:

b1. RAM van #9800 totaan #9FFF.

b2. RAM vanaf #4000 en dan liefst zoveel mogelijk.

b3. joystickaansluiting.

- -

ad.1 ook deze kunnen gestapeld worden.

ad.2 spreekt voor zich. Portemonnee is belangrijkste factor.

ad.3 volgens de clubstandaard. Zie AcornNieuws '82 nr.5.

In een later stadium zullen we het ook nog hebben over printer, DOS, 80kolommen (dit soort zaken wordt voorlopig niet, wellicht zelfs nooit "gestandaardiseerd", want we willen niemand op kosten jagen. Deze items zullen worden behandeld in de trant van: ALS u een printer etc. heeft, DAN wordt u geacht deze als volgt aan te sluiten), systeemroutines, schakelbits, decoderingen etc. etc.
Tot zover deze eerste aanzet. We beseffen dat het nauwelijks spectaculair is, maar het moest gewoon een keer gebeuren. Denk ook niet dat deze goede bedoelingen van hogerhand zonder meer vruchten zullen afwerpen. We zullen met z'n allen moeten coöpereren om e.e.a. te laten werken. Het is een samenspel van federatie, redactie EN de leden.

namens de hardware commissie, Bram Poot

P.S. Als service aan het legertje nieuwe leden van afgelopen jaar wordt elders in dit blad nogmaals beschreven hoe het zgn. stapelen van IC's in z'n werk gaat. Voor de overige verwijzingen kun je wellicht terecht bij je medeleden, drukwerkarchief en/of programma-archief van je regio. Als dat onverhoopt niet mocht lukken, kun je altijd nog de hardware commissie aanschrijven.

No matter how definite a standard is,

IBM can always redefine it.

S T A P E L E N

Om de geheugenruimte van de computer te vergroten kunnen we een aantal RAM chips (2114) letterlijk op elkaar stapelen.
We kunnen zo nog 1K geheugen erbij plakken in het lage geheugen (#3C00 - #3FFF), en 2K in het hoge geheugen (#9800 - #9FFF).
We zetten de ene 2114 op de andere. Dan zitten de potjes van de bovensta al enigsinds 'klemmend' over die van de onderste. let op de pennen 1, deze zitten boven elkaar!
We solderen de pootjes van de bovenste nu vast op de onderste, alleen pootje 8 buigt u ietsje opzij. Dit is nl. de "chip select poot", die naar een dekoder moet.
U steekt nu de IC's 10 en 11, ieder voorzien van een bovenbuurman (vrouw) weer terug in de sockets. De twee pennen 8 verbindt u met elkaar, en met pen 7 van IC 6.
Nu is het lage gedeelte klaar
Met IC 32 en IC 33 doet U hetzelfde, alleen nu verbindt u de beide pootjes 8 met pen 7 van IC 30
Tenslotte ook met IC 34 en IC 35, en deze vrije pennen 8 aan pen 9 van IC 30.

Nu heeft u er van #3C00 tot #4000, en van #9800 tot #A000 geheugen bij gekregen. U bezit nu een atom met laag 6K, en hoog 8K geheugen.
Vanwege de bouwwijze wordt deze uitbreidig "gestapeld geheugen" genoemd.

Acorn Nieuws 1982, Bewerkt: R. van Drunen.

****
PROBLEMEN BIJ GESTAPELD GEHEUGEN
****

Øvergenomen uit het computerclubblad Overijssel/gelderland Jrg.1 no. 1

Heeft u in de computer zelf geheugen gestapeld?
Dan is het verstandig om de chip selectie lijntjes van IC 6 (74LS138), welke naar de gestapelde rams gaan, ook naar IC5 (74LS30) te voeren.
Zijn deze verbindingen niet aanwezig, dan kunnen er in samenhang met de schakelkaart moeilijkheden ontstaan.
Zo kon ik soms #3C00 t/m #3FFF niet voor de tekst-editor gebruiken.

Licht hiervoor van IC5 pin 1 uit het voetje en verbindt deze met een draadje met IC6 pin 7. (Dit voor het geval er alleen van #3C00 t/m #3FFF gestapeld is) Nu gaat het wel goed.

Heeft U ook van #2000 t/m #23FF en van #2400 t/m #27FF gestapeld, dan zullen er nog weer een paar diodes en een weerstand bij moeten komen.
Aan elk van de pootjes IC6 pin 7 (i.p.v. het draadje van bovenstaande), IC6 pin 14 en IC6 pin 15 komt een diode, met de kathode naar het pootje. De anodes komen samen en gaan naar IC5 pin 1 (deze mag dus niet meer aan +5 volt liggen). Ook gaat er van IC5 pin 1 een weerstand van 1Kohm naar +5 volt. U bent nu verzekerd van een goede werking van deze geheugen gebieden in samenwerking met de schakelkaart.

Veel succes met het ombouwen toegewenst,

Wibo Visser
Emmeloord.

OMBOUW SCHAKELKAART

Als U net als mij de schakelkaart bezit, maar geen geheugenkaart en U bent eigenlijk niet van plan om laatstgenoemde (binnen afzienbare tijd) aan te schaffen, dan is dit artikel wellicht interessant voor U.

Ik had de C-MOS RAM's op zowel A0 als op E0 gestapeld (A1 en E1 dus), maar gebruikte het E1 blok eigenlijk niet, omdat ik mijn SOS op E0 heb. Doodzonde van de dure RAM chips. Ik vond dat hier maar eens een einde aan moest komen. Die 4k kan nuttiger gebruikt worden!

LET WEL, DIT ARTIKEL BESCHRIJFT DE OMBOUW VAN EEN SCHAKELKAART OUD MODEL.

Nu zouden er voor mij 2 mogelijke bestemmingen zijn:
- -de 4k naar #1000.
- -de 4k naar #4000.

De eerste keuze biedt de mogelijkheid om P-CHARME op #1000 te zetten. Of om hier een SOS te plaatsen (de floating point moet dan wel (weer) aangepast worden).
De tweede keuze geeft een verlengde lower text-space, makkelijk voor super lange programma's of voor lange teksten op de tekst-editor. Het is hierbij natuurlijk wel noodzakelijk, dat er in de komputer al geheugen gestapeld is van #3C00 to #3FFF.

Bij een verandering als deze streeft men er altijd naar om met zo weinig mogelijk ingrepen, zo veel mogelijk te bereiken. Zo ook in dit geval. Ik hoop dat ik er in geslaagd ben. Overigens is het geheel natuurlijk ook uit te voeren, als U nog niet gestapeld hebt, maar dan moet U dat alsnog wel even doen.

Ik ben van het schema van de schakelkaart uitgegaan, zoals zich dat op blz. 50 en 51 van A.N. 1 jaargang 3 bevindt.

Er moet in principe op 4 plaatsen ingegrepen worden:
- -het 'openen' van het DATA lijnen buffer (de LS245), als het verschoven E1 blok aangesproken moet worden. Dit geldt natuurlijk zowel voor schrijf- als voor lees operaties.
- -de selektie logika, welke de RAM's en de EPROM's selekteerd.
- -de mogelijkheid om vrij in de RAM te kunnen schrijven, terwijl de rest toch nog achter de Write Protect zit. Dit is dus eigenlijk alleen nodig als de verschuiving naar #4000 plaats vindt.
- -de RAM chips zelf.

De realisatie (proloog).

Alvorens de realisatie te bespreken, wil ik nog graag kwijt dat ik bewust voor een oplossing zonder extra chips heb gekozen, omdat er anders een onhoudbare situatie zou ontstaan wat betreft het aan elkaar plakken en hangen van IC pootjes. Ik heb daarom voor een oplossing met dioden en weerstanden gekozen. De gebruikte dioden zijn van het type 1N4148 of equivalent en de

weerstanden zijn 1/8 Watt groot. Met de weerstanden en de diodes worden de eenvoudigste digitale poorten opgebouwd die maar mogelijk zijn en het geheel is redelijk plat te houden. Zo wordt er niets overbodigs aangebracht en de onderdelen zijn goedkoop en makkelijk verkrijgbaar (in principe zou iedere supermarkt ze moeten hebben).

De realisatie (finale).

Om maar met het eerste punt te beginnen: het DATA lijnen buffer (de LS245) werkt alleen als er een adres aangesproken wordt dat betrekking op de geheugenkaart heeft. Het betreft dus de adressen #A000 t/m #AFFF, #E000 t/m #EFFF en #BFFF. We willen nu het geheugen gebied #1000 t/m #1FFF cq #4000 t/m #4FFF aan de kaart toevoegen. Voor het gekozen gebied moet het DATA buffer nu ook open gaan, om DATA door te laten. Hiertoe wordt pootje 19 van de LS245 laag gemaakt (boven EN -van enable- moet in het schema nog een streepje staan, omdat deze ingang laag aktief is). Dit pootje wordt laag gemaakt door IC1 (74LS10). Linksboven in het schema vindt de detektie van A0, E0 en #BFFF plaats. Dit moet dus uitgebreid worden met 10 resp. 40. Dit vormt een OR funktie, zodat we met dioden en een weerstand een OR poort moeten maken. Deze OR poort maken we als volgt. De verbinding tussen IC 1 pootje 12 (vanaf hier aangegeven met 1-12) en 1-11 moet doorgekrast worden. Vanaf 1-11 leggen we een weerstand van 33kOhm naar massa (weerstand niet kleiner nemen!). Tussen 1-12 en 1-11 komt een diode, met de anode (de kant waar het streepje NIET zit) aan 1-12. Verder komt er een diode van 5-10 (de 4001) naar 1-11 (deze is voor het verschoven E1 blok). Er wordt overigens juist op dit punt in het schema ingegrepen, omdat de ENABLE ingang van het buffer geklokt is (1-10).

De selektie logika voor het selekteren van de RAM's en de EPROM's wordt als volgt aangepast.
In het geval dat er voor een verschuiving naar #4000 gekozen wordt, moet er het volgende gebeuren. Aan de bestukte kant (bovenkant) van de print wordt de verbinding tussen 7-10 (74LS138) en 6-9 (74LS32), zo dicht mogelijk bij 6-9, doorgekrast. Ook wordt aan de bovenkant van de print, het printspoor, dat op 2-1 (74LS14) uitkomt, doorgekrast (er hoeft toch geen selektie meer plaats te vinden tussen E0 en E1). Vanaf 6-9 komt een weerstand van 33kOhm naar massa. Vanaf de bovenste LS244, in het schema, komen diodes op de pootjes 18,16 en 5, met de anode aan de genoemde pootjes en de kathode's samengevoegd aan 6-9. De bedoelde LS244, is op de print het IC, dat niet in de midden zit. Tenslotte komt er een draadje van pootje 3 van dezelfde LS244 naar 2-1.
Voor de verhuizing naar #1000 gaan we ook als hierboven omschreven te werk. De diodes komen echter op de pootjes 3,16 en 18 van de LS244 te zitten en pootje 5 gaat naar 2-1.

Voeren we de verhuizing naar #4000 uit, dan zullen we de RAM op #4000 niet achter de write protect willen hebben. De Write Protect zal dus gescheiden moeten worden. We zullen de blokken A0 en E0 achter de W.P. willen houden. Via 1-6 wordt het schrijven van DATA in de RAM geregeld. Dit schrijven moet geklokt gebeuren (1-5). Het zal duidelijk zijn, dat we 1-6 niet meer kunnen gebruiken. We moeten zelf een NAND poort moeten

maken. Met diodes en weerstanden kan geen NAND poort gemaakt worden, maar gelukkig is er nog een inverter beschikbaar in IC2. Hiervoor moeten we wel het printbaantje van 2-11 naar 1-3 doorkrassen. Nu leggen we aan 1-4 en 1-5 een diode met de kathode naar de genoemde punten. De anode's van deze diode's gaan samen naar 2-11. Verder komt er een weerstand van 33kOhm van 2-11 naar +5Volt. Nu wordt 2-10 met pootje 21 van de beide 6116 chips verbonden, welke het nieuwe blok vormen. Deze pootjes moeten hiervoor los van de eventueel boven- of onderliggende 6116 zijn.

Hiermee is het laatste punt eigenlijk ook al besproken. Zij die al gestapeld hebben, moeten pootje 21 van beide 6116's even loshalen. En zij die nog niet gestapeld hebben, stapelen door de 6116 op elkaar te solderen, op de pootjes 18 en 21 na.

Als alles zoals hiervoor beschreven is uitgevoerd, moet het werken.

De uitvoering (toegift).
========================

Als laatste noot nog het volgende: heeft u in de komputer zelf 2114's gestapeld? Dan is het verstandig om de selektie lijntjes van IC 6 (LS138), welke naar de gestapelde RAM gaan, ook naar IC5 te voeren. Zijn deze verbindingen niet aanwezig, dan kunnen er in samenhang met de schakelkaart moeilijkheden ontstaan. Zo kon ik soms #3C00 t/m #3FFF niet voor de tekst-editor gebruiken.

Licht hiervoor 5-1 van het moederbord uit de voet en verbindt deze dmv een draadje met 6-7 (dit voor het geval er alleen van #3C00 t/m #3FFF gestapeld is). Nu gaat het wel goed.

Heeft U ook #2000 t/m #23FF en #2400 t/m #27FF gestapeld, dan zullen er nog weer een paar diodes en een weerstand bij moeten komen. Aan elk van de pootjes 6-7 (ipv het draadje van bovenstaande), 6-14 en 6-15 komt een diode, met de kathode naar het pootje. De anodes komen samen en gaan naar 5-1 (deze mag dus niet meer aan +5Volt liggen). Ook gaat er van 5-1 een weerstand van 1kOhm naar +5Volt. U bent nu verzekerd van een goede werking van deze geheugen gebieden in samenwerking met de schakelkaart.

Veel succes met het ombouwen toegewenst,

Wibo Visser
Emmeloord

****
** CASSETTE-PROBLEMEN EN MOGELIJKHEDEN **
****

Ofschoon reeds veel over dit onderwerp is gepubliceerd, zijn er nog steeds nieuwe ontwikkelingen die de moeite van het publiceren waard zijn, en zijn er ook nog steeds "beginners", voor wie de oude publikaties onbekend of niet meer toegankelijk zijn.

Vooral voor beginners die weinig geld willen uitgeven voor opslag van programmatuur is de cassette nog steeds het meest geschikte medium.
Daarbij is de verleiding groot, dan ook gebruik te maken van een of andere simpel cassette-apparaat; dat kan best goed gaan, maar ook voor moeilijk op te sporen problemen zorgen.
Een duurder of zelfs een heel duur cassette-apparaat kan echter ook voor problemen zorgen.
Zoals wellicht bekend, kan met de Atom gewerkt worden op verschillende snelheden.
Het systeem werkt standaard op 300 baud, een snelheid die nog het minst critisch is, maar toch ook nog problemen kan opleveren.
Daarnaast is veel software op 1200 baud opgeborgen, hetgeen al heel wat sneller is, maar ook critischer.
Hiervoor is het in ieder geval nodig, in de Atom op te nemen:

- ofwel het ALLCOS programma;
- ofwel JOSBOX, P-CHARME, ASBK-I of soortgelijke Eproms in de IC 24 voet of op een schakelkaart.

Ook kan de Atom bedreven worden op 4800, 6000 en zelfs 9600 baud; vooralsnog gaat daarbij de voorkeur naar 4800 baud; dat is al behoorlijk snel.
Het spreekt vanzelf dat hiervoor een echte goede cassette recorder nodig is, maar ook speciale software genaamd MJCOS, en een kleine wijziging aan de Atom.

Publicaties over deze onderwerpen zijn te vinden in de jaargang '82 (samenvatting), '83, en deels 1984.
Daarover alsmede over allerlei problemen die bij het laden en saven te voorschijn kunnen komen, en die meer liggen op het terrein van de software, zal in een volgende publicatie het een en ander gezegd worden.
Op dit moment zouden wij bij wijze van uitzondering alleen alvast willen wijzen op het volgende.
Momenteel zijn van de hiervoor genoemde Eproms (JOSBOX etc.)nieuwe versies verkrijgbaar waarin een routine zit die met name de 1200 baud snelheid veel minder critisch maakt; moet je er dus nog een aanschaffen, vraag dan daarna.
In deze aflevering willen we vooral aandacht schenken aan de maatregelen op "hardware" bebied en hetgeen daar direct mee te maken heeft.

Waarom gaat het:
================

a. Het signaal van de Atom behoorlijk op de cassette te zetten.

b. Het signaal van de cassette zodanig te "maken" dat de Atom het zonder fouten kan verwerken.

HET SAVEN

Wat gaat hier zoal fout:

- de signalen worden door toerenregeling en correctienetwerken in de recorder zodanig vervormd, dat van de oorspronkelijke blokgolf weinig overblijft,
- ofwel de automatische volume regeling staat te laag ingesteld.

Dit laatste kan goed gecontroleerd worden door een bandje af te spelen op een andere cassette-recorder die een (in- en) output aanwijzing heeft; als het goed is, komt de aanwijzing op 0 dB.Haalt de output niet meer als 7 - 10 dB dan moet een bevriend elektroniker in de arm genomen worden om de recorder bij te stellen, ofwel proberen de werkwijze die we straks in hoofdstuk "Het LADEN" zullen beschrijven.
Of de blokgolf teveel vervormd wordt, kan eigenlijk alleen goed gemeten worden met een scoop.
Om ervoor te zorgen dat het signaal goed op de recorder komt kan gebruik worden gemaakt van de, in Acorn Nieuws No.4 1982 beschreven, filterschakeling of in No.2 jaargang 2 (1983) blz. 19 - 23.

HET LADEN.

In de dagelijkse praktijk werken we welliswaar vaak met bandjes welke op de eigen recorder zijn opgenomen, maar ook zeer frequent met bandjes die gekocht of van anderen zijn overgenomen.
En dan is, de hievoor in het hoofdstuk "SAVEN" genoemde werkwijze uiteraard weinig effectief.
Daarom dan ook speciale aandacht. Allereerst een weinig bekende verbetering de condensator C7, te vinden schuin links "boven" IC23,dient een waarde te hebben van 100 nF; Het heeft zin deze te controleren. Volgens berichten zijn er Atoms waarin C7 deze waarde reeds heeft, zoniet, is het aantrekkelijk de 100 nF condensator te installeren; zelfs de 300 baud input wordt daardoor minder critisch.

2.Een meer algemene verbetering wordt bereikt door een speciale " INPUT-INTERFACE-SCHAKELING" die versterkers en filters bevat, zoals weergegeven in het schema.
Het idee voor deze schakeling is afkomstig van de BBC-ingangsschakeing, die dan geheel aangepast is aan de Atom behoeftes en mogelijkheden.
Het resultaat:
In combinatie met de moderne 1200 baud routines, levert deze interface over een zeer brede range van volume-setting en signaalvorm een foutloze input aan de Atom. (Regelen tussen stand 1 en 8 gaf zelfs tijdens het laden geen errors).
Het voorddeel is dus, dat nu ook "moeilijke" cassettes gelezen kunnen worden, vaak zelfs met fout staande koppen.
Proeven met MJCOS op 4800 baud gaven ook prima resultaten, ook met eenvoudige recorders;
zelfs op 9600 baud is het signaal op de scoop uitstekend.
Voor de 4800 baud en 9600 baud is het nodig enkele condensators aan te passen, aan een definitieve versie voor deze snelheden, eventueel omschakelbaar wordt nog gewerkt, U hoort er nog van.

Erg aantrekkelijk is het deze INPUT-INTERFACE in de Atom op te nemen, hij kan echter ook in een apart "tussendoosje" in de

kabel.
Inbouw in de Atom heeft overigens als negatieve consequentie dat er problemen ontstaan als de cassette-inputpoort wordt gebruikt voor andere doeleinden.(bv. LF-counter e.a.).

3.Ondanks de goede diensten van de input-interface, kan het toch nog misgaan met het laden.
Er zijn dan grofweg nog twee fouten:

- De kop in de recorder staat anders dan in de recorder waarop de band was opgenomen.
  Bijstellen is een secuur werkje, waarvoor echter moet worden uitgegaan van een cassette die is opgenomen op een erkend goede recorder, of nog beter met een echte testcassette.
- de cassette is met een andere snelheid opgenomen.

Om deze en andere problemen op het spoor te komen zijn 2 programma's, waarvan de listings zijn bijgevoegd.
Het programma "Recorder diagnose" laat de signaalvorm zien, en kan ook de frequentie meten.
Het programma "Tape speed" is voor dit laatste doel echer veel beter.Daarmee wordt namelijk zeer duidelijk de afwijking van de band op basis v.d. "leader" weergegeven.
Ook hier weer: als men de snelheid van de eigen recorder wil bijstellen doe dit met een cassette waarop op een erkend goede recorder de leader-toon geruime tijd is opgenomen.

Henk Bastings.
Redactie: B.Tossaint.

```
  1 REM TAPE SPEED TESTER
 10 DIMPP(1)
 20 P.$21;GOS.a;GOS.a;P.$6;@=0
 30 LINK#FC4F;REM PRINT PLAY TAPE
 40bLINK#FB8A;REM WAIT 1/2 SEC
 50 ?#80=0;?#81=0;X=56;REM PRESET COUNTS
 60 LINKPP0
 70 P."TAPE IS"ABS((?#81)-200)/2,"%"
 80 IF?#81>200 P."FAST"';G.b
 90 P."SLOW"';G.b
100 E.
110aDIMP-1
120[
130:PP0 LDA#B002
140 TAY
150 EOR#80;\PREVIOUS STATE OF #B002
160 STY#80;\SAVE NEW STATE
170 TAY
180 AND@#20;\TAPE SIGNAL SAME?
190 BEQ PP1
200 INC#81;\IF TAPE INPUT CHANGE
210:PP1 TYA
220 AND@#10;\ OSC 2.4 KHZ CHANGE
230 BEQ PP0
240 INX \IF 2.4 TRANSITION
250 BNE PP0\ UNTIL 200 TRANSITIONS
260 RTS
270]
280R.
```

```
  1 REM RECORDER DIAGNOSE
 10 DIMB(255),LL(6),P(-1)
 15 F.I=0 TO 6;LLI=P
 20 P.$21;GOS.a;P.$6$12
 30 IN."SIGNAAL OF FREQUENTIE"$B
 40 P.''"PLAY TAPE"'
 50 IF ?B=CH"S";G.s
100 IN."METEN ROND FREQ. (BV 2400)"F
110 P.$12,"FREQUENTIE:";@=5
120 DO LINK LL0
130 T=0;S=0;E=(500000/F-37)/17+1
140 L=7*E/10;H=10*E/7
150 F.K=0 TO 255
160 IF B?K>L;IF B?K<H;S=S+B?K;T=T+1
170 N.K
180 G=500000*T/(17*S+20*T)
190 P.G,$8$8$8$8$8
200 UNTIL 0
300sP."DRUK OP TOETS VOOR MONSTER"'
305 F.K=1 TO 60; WAIT; N.
310 CLEAR4
320 F.K=10 TO 180 S.20
330 LINK#FFE3;LINK LL5
340 MOVE(-1),K
350 F.I=0 TO 255
355 J=B?K+K
360 DRAWI,J;MOVE(I+1),J
370 N.;N.;E.
400a[\TIJDMETING EN SAMPLE-ROUT
405:LL0 LDX@0
410:LL1 LDA#B002;STA#80
420:LL2 LDA#80;EOR#B002;AND@#20
430 BEQ LL2 \WACHT OP FASE OM
440:LL3 LDY@0;LDA#B002;STA#80
450:LL4 INY;BEQ LL1;LDA#80
460EOR#B002;AND@#20;BEQ LL4
470 TYA;STAB,X;INX;BNE LL3;RTS
500:LL5 LDX@0
510:LL6 LDA#B002;AND@#20;LSRA
520 LSRA;STA B,X;INX;BNE LL6
530 RTS;];R.
```

****
MJCOS & PROGRAMMER TROUBLES
****

Reeds enige tijd werkte ik met de Acorn club-programmer tot volle tevredenheid. Op een slecht moment was er een slimmerik die MJCOS uitvond.
Omdat ook ik mijn recorder graag wilde "opvoeren" werd de Acorn hard- en soft-warematig aangepast. Het shiftout-bitje (bitje 5) in poort A van de VIA werd verbonden met de DIN-plug van de cassette in/uitgang. Alles werkte perfect.
Dacht ik.
Totdat ik weer eens Eproms moest programmeren. Na 2 Eproms geprogrameerd te hebben, en waarbij "bergen" verify-errors zaten, kwam ik op 't idee dat de programmer wel eens kapot kon zijn. Na lang zoeken en prutsen geen resultaat: de programmer was okee.
De software dan? Zelfde resultaat: geen dus.
Plotsklaps was er ineens het lunimeuze idee: de laatste wijziging (tijden geleden) was de MJCOS hardware aanpassing. De signalen op de DIN-plug kwamen ook op de VIA poort.
De oplossing voor dit probleem is eenvoudig: behalve de weerstand die gelegd moet worden tussen de VIA en de DIN-plug, prutsen we er ook nog een diode bij.
Deze diode (de goedkoopste die je kunt krijgen bv. 1n....) komt in serie met de weerstand, zodanig, dat het streepje (kathode) naar de 6522 poort "wijst". De 6522 kan nu wel de recorder "laag" trekken, maar de recorder niet de VIA. Aldus kunnen zowel MJCOS als de programmer in gebruik blijven, zonder schakelaars of soldeerbouten.

Gerhard Visser &
Evert van Schothorst.

Dynamische RAM kaart

TELEC Groningen verkoopt 32 kByte dynamische RAM kaarten voor de Acorn Atom, voor een m.i. redelijke prijs. De informatie die bijgeleverd wordt is echter zeer summier. Een schema schijnt niet leverbaar te zijn.
Zo moet je zelf nog wat in de komputer toevoegen om niet in de problemen te komen. Bekijken we het schema van ons raspaardje. TELEC zegt: verwijder IC 6. Goed, dat is logisch, want er is geen selektie meer nodig voor de IC's 10 t\m 19. Simpelweg omdat deze niet meer in de komputer zitten (haal de 2114's er ook echt uit, om andere narigheden te voorkomen).
Verwijderen we IC 6, dan komen pootjes 2,3,4,5 en 6 van IC 5 (774LS30) los te hangen; foute boel. Er kunnen vreemde dingen gebeuren. Die traden dan af en toe ook op bij mij.
Wat gaan we er aan doen? We verwijderen inderdaad IC 6. Maar dan moeten we om de zaken netjes te houden, de genoemde ingangen van IC 5 wel even met wat weerstandjes aan de +5 Volt knopen (hierdoor worden deze ingangen inaktief). Dan wordt het DATA buffer (IC4 8304) tenminste weer goed gestuurd. Een weerstandwaarde van rond de 2 kOhm voldoet uitstekend.
Wat we ook kunnen doen, is IC 6 laten zitten en pootje 6 van dit IC aan massa leggen. Kras hiervoor de baan die op pootje 6 uitkomt door en verbindt het pootje met een draadje naar massa (pootje 8 van hetzelfde IC).

De 2de methode is wel wat ingrijpender dan de 1ste, maar werkt net zo goed. De eerste methode verdient mijn voorkeur.
Met deze toevoeging werkt de kaart bij mij geheel zonder problemen. En dat is natuurlijk wat we nastreven.

Wibo Visser

OPROEP

Wie heeft een originele 32 kByte dynamische RAM-kaart van Acorn in gebruik en deze met een AB-connector buiten de bus gebracht.
Graag een berichtje naar de redaktie met beschrijving van eventuele aanpassingen.

Ombouw programmer.

Omdat ik laatst een stel 2732's tegen een zeer aantrekkelijke prijs over kon nemen, besloot ik m'n programmer hier ook maar geschikt voor te maken ...
Bestudering van Acorntjes Brood 2.4 leerde dat het niet moeilijk kon zijn. Alleen die oplossing met al die losse schakelaars (wie heeft er nu een drie-polig omschakelaar van redelijk formaat in de kast liggen?), zag ik niet zo zitten.

Dan zelf het probleem nog maar eens bestuderen. De 2732 en de 2532 verschillen kwa aansluitingen op drie punten. Het betreft de aansluitpennen 18, 20 en 21. 2532 plaatst hier achtereen- volgens A11, CE en Vpp, terwijl 2732 het met CE, OE/Vpp en A11 doet.
Dit naast het feit dat er in principe 6 (zes) mogelijke situaties ontstaan, te weten:

- 2532 lezen (25R)
- 2532 programmeren 21V (25P21)
- 2532 programmeren 25V (25P25)
- 2732 lezen (27R)
- 2732 programmeren 21V (27P21)
- 2732 programmeren 25V (27P25)

Uit deze gegevens laat zich eenvoudig een aansluittabel voor beide types EPROM en alle situaties distilleren. Deze kan er dan als volgt uit zien:

pen \	25R	25P21	25P25	27R	27P21	27P25
18 \	A11	A11	A11	CE	CE	CE
20 \	CE	CE	CE	low	21V	25V
21 \	5V	21V	25V	A11	A11	A11
stand	1	2	3	4	5	6

En wat is er nu makkelijker om het geheel omschakelbaar te maken met een enkele schakelaar (alles hardware matig zo ombouwen, dat het software matig te schakelen is, hoor ik iemand zeggen ...). De keus valt hiermee op een draaischake- laar. En wel een met minimaal zes standen en minimaal 3 p-kontakten.

Nu weten we dus exact wat we op de pennen 18, 20 en 21 aan moeten bieden, om het beoogde te bereiken. Voor de ombouw krassen we de koperbanen op de programmer print vlak bij het EPROM voetje door bij de genoemde pennen. Aan de genoemde pennen komen nu de moeder (p) kontakten van de draaischakelaar. De schakelkontakten worden dan volgens de tabel aangesloten.

Tip: koop bij de draaischakelaar een knop met een standenaan-duiding, zodat steeds duidelijk blijft in welke stand de programmer staat.

Wibo Visser.

LOGIKA.

Onder deze noemer zal ik trachten in de komende info blaadjes van onze regio het een en ander uit de doeken te doen over allerlei digitale toestanden. Het idee hierachter is, dat er veel nieuwe leden zijn, die niet helemaal (cq helemaal niet) op de hoogte zijn van het digitale gebeuren rondom de computer. Hier zullen we dus trachten wat verandering in te brengen. Tevens zullen de artikelen als "naslagwerk" voor de meer gevorderden onder ons van dienst kunnen zijn.
Om maar "makkelijk" te beginnen deze keer iets over eenvoudige digitale (hier kombinatorische) bouwstenen. In eerste instantie zullen we 4 basis elementen gaan onderscheiden.

Inleiding.

In de wereld van de digitale elektronika draait alles om 2 nivo's (toestanden):

laag	hoog
0 Volt	5 Volt
0	1

In eerste instantie kunnen we zeggen, dat de gehele logika op deze 2 nivo's gebaseerd is. Alleen met deze 2 nivo's zijn we er natuurlijk niet. Daar moet iets mee gebeuren. Aan de hand van deze 2 nivo's gaan we bewerkingen uitvoeren. Met deze bewerkingen kunnen we komplexe logische funkties realiseren. Voor het zover is zullen we eerst de eenvoudige bewerkingen bekijken.

AND-gate.

In de eerste plaats is daar de AND-gate (EN poort). Bekijken we de AND-gate met 2 ingangen, dan kunnen we deze vergelijken met 2 schakelaars in serie. Slechts als beide schakelaars gesloten zijn (wat overeenkomt met "hoog") zal het ingangssignaal aan de uitgang beschikbaar zijn. Figuur 1 geeft het Amerikaanse (Texas Instruments) symbool van de AND-gate weer. Figuur 2 het IEC symbool (IEC is de nieuwe Europese norm). Voor elke poort kunnen we een tabel opstellen, welke de funktie van de poort weergeeft. Terecht wordt deze tabel de "funktietabel" genoemd. Als we de 2 ingangen van de gate x resp. y noemen en de uitgang noemen we S, dan kunnen we voor de AND-gate de volgende funktietabel opstellen:

x	y	S
0	0	0
0	1	0
1	0	0
1	1	1

We lezen deze funktietabel als volgt. Als de ingangen x en y beide laag (0) zijn, dan is de uitgang S ook laag. Als vervolgens x laag is en y is hoog, dan blijft S laag. Is x hoog

en y laag, dan is S nog steeds laag. Tenslotte als x en y beide hoog zijn, dan is S hoog (vergelijk de situatie bij de 2 schakelaars in serie, waarbij hoog voor dicht staat en laag voor open). Twee voorbeelden van een AND-gate zijn de 74LS08 uit de TTL serie en de 4081 uit de CMOS of 4000 serie.

INVERTER.

Vervolgens zullen we de INVERTER bekijken. Een Nederlandse naam zou NIET poort kunnen zijn. De naam zegt het misschien al. De uigang is niet gelijk aan de ingang, d.w.z. als de ingang laag is, is de uitgang hoog en omgekeerd. De INVERTER heeft maar 1 ingang en 1 uitgang. De funktietabel ziet er dan als volgt uit:

x	S
0	1
1	0

Figuur 3 geeft het symbool volgens de Amerikaanse notatie weer en figuur 4 volgens de IEC norm. Het kleine cirkeltje wordt inversie cirkel genoemd en duidt de inverterende werking van de poort aan. Voorbeelden van de INVERTER zijn de 74LS14 en de 4049.

Als we nu een NIET poort achter een EN poort zouden plaatsen zouden we een NIET EN poort krijgen (logisch, of niet soms). Deze poort bestaat inderdaad ook en wordt NAND-gate genoemd. De funktietabel van de NAND-gate zal dus het omgekeerde beeld geven voor de uitgang S t.o.v. de AND gate. Iedereen kan die dus voor zich opstellen. Het Amerikaanse symbool staat in figuur 5 en het IEC symbool in figuur 6. Als NAND-gate kunnen de 74LS00 en de 4011 genoemd worden.

OR-gate (inclusive).

Als derde poort zullen we nu de OR-gate beschouwen. Het betreft hier de zgn inclusive-OR, maar hierover straks meer. Deze OR-gate is te vergelijken met 2 schakelaars die parallel staan. Zodra een van de schakelaars gesloten is, zal het ingangssignaal aan de uitgang verschijnen. De funktietabel is dan als volgt:

x	y	S
0	0	0
0	1	1
1	0	1
1	1	1

We zien dat als zowel x als y hoog zijn, S ook hoog is (vergelijk beide schakelaars gesloten). Hierin zit het "inclusive": als zowel x als y hoog is, is S ook hoog.
Figuur 7 geeft het Amerikaanse symbool van de inclusive OR-gate, figuur 8 geeft het IEC symbool. Voorbeelden van de OR-gate zijn de 74LS32 en de 4071.
Ook hier kunnen we weer een INVERTER achter plaatsen. Dan krijgen we een NOR-gate. De funktietabel is voor S dan dus net omgekeerd als bij de OR-gate. De figuren 9 en 10 geven resp.

het amerikaase en de IEC symbool weer (let ook weer op de inversie cirkel). Voorbeelden van de OR-gate zijn de 74LS32 en de 4071. Zij nog vermeld, dat de inclusive OR-gate in het dagelijks leven gewoon OR-gate wordt genoemd.

OR-gate (exclusive).

Als vierde en voorlopig als laatste beschouwen we de exclusive OR-gate. Deze is gelijk aan de inclusive OR-gate. Het enige verschil is dat als zowel x als y hoog zijn, uitgang S nu laag is (exklusief). In de figuren 11 en 12 staan het Amerikaanse en het IEC symbool. Voorbeelden zijn de 74LS86 en de 4030.
Ook hier kunnen we weer een INVERTER plaatsen. Zie de figuren 13 en 14. Voorbeeld is de 74LS266.
Normaal gesproken wordt de exclusive OR-gate aangeduid met EXOR.

FINALE.

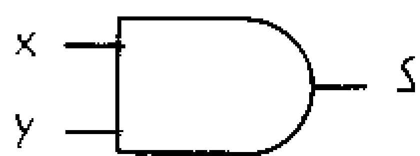
Nu we een aantal "basis" vormen van digitale bouwstenen hebben bekeken, kunnen we ons afvragen, wat we er mee kunnen. Hier over een volgende keer meer.
Tot slot nog een overzicht van de behandelde poorten:

x y	AND	NAND	OR	NOR	EXOR	EXNOR
0 0	0	1	0	1	0	1
0 1	0	1	1	0	1	0
1 0	0	1	1	0	1	0
1 1	1	0	1	0	0	1

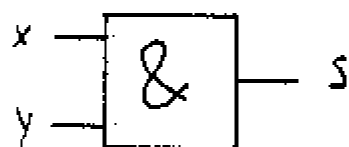
x	NOT
0	1
1	0

In woorden:
AND S = 1 als x=1 en y=1
NAND S = niet 1 als x=1 en y=1
OR S = 1 als 1 of meer ingangen 1 zijn
NOR S = niet 1 als 1 of meer ingangen 1 zijn
EXOR S = 1 als uitsluitend 1 ingang 1 is
EXNOR S = niet 1 als uitsluitend 1 ingang 1 is
NOT S = inversre van x

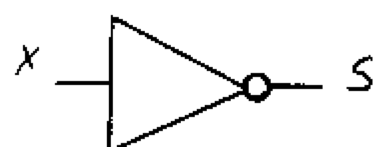
logische groeten,
Wibo Visser.



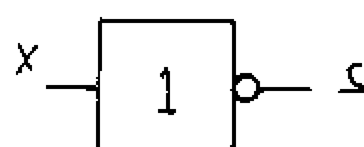
AND FIG. 1



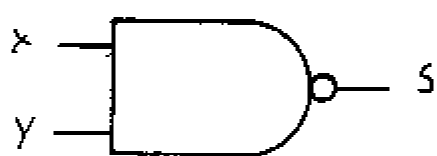
AND FIG. 2



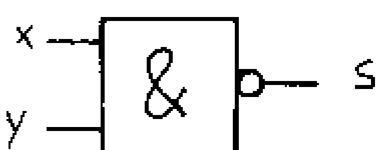
NOT FIG. 3



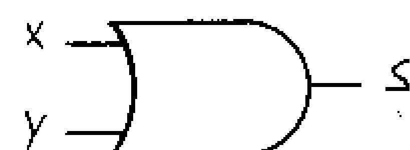
NOT FIG. 4



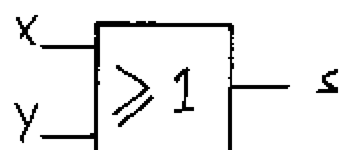
NAND FIG. 5



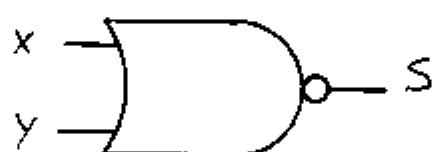
NAND FIG. 6



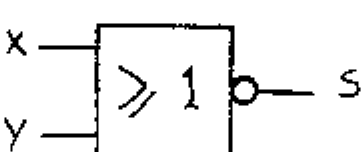
OR FIG. 7



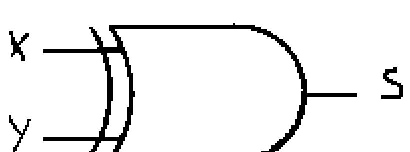
OR FIG 8



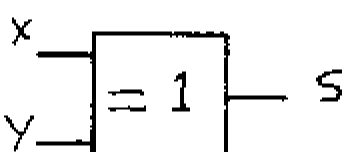
NOR FIG. 9



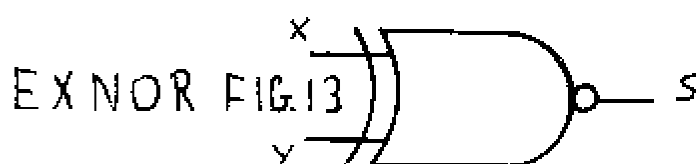
NOR FIG. 10



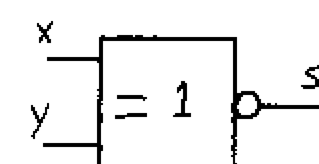
EXOR FIG. 11



EXOR FIG. 12



EXNOR FIG. 13



EXNOR FIG. 14

In Acorn Nieuws no.3 van dit jaar staat een artikel van Hans Heuts genaamd 'Hulpjes'.
Wat betreft de z.g. snellere goto's ontdekte ik toevallig dat het machine-code gedeelte kan vervallen.
Voor alle duidelijkheid staat hieronder een praktijkvoorbeeld.
(n.b. exakt intikken i.v.m. de statementpointer)

```
10A=#2907;B=A
20P."1 ";P."2 ";P."3 ";P."4 ";P."5 ";P."6 ";P."7 ";P."8 "
30IFA>#293C;A=B+7;B=A
40IFB>#293C;G.10
50A=A+7;!5=A
99E.
```

Als je dit programma intikt, runt en bekijkt zal je bemerken, dat met het 'poken' van een geheugenadres in de statement-pointer het programma verder zal gaan op het adres wat ingebracht is.
Het enige waar je op moet letten is dat de statement-pointer naar een regelovergang (#0D) of naar een ';' (#3B) wijst, anders krijgt je ATOM last van ingewandsstoornissen.
Indien je in tekstpagina #29 werkt zal !5=#2900 hetzelfde zijn als run.

Je kan dit kommando dan ook in een programma of in direkt-mode gebruiken, dit betekend uiteraard dat je midden in een programma-regel kan beginnen (oppassen voor FOR-NEXT, DO-UNTIL etc.)

Willem Kautz

LOGISCHE SIMULATOR VOOR DIGITALE SCHAKELINGEN

1. Inleiding

Dit artikel beschrijft een programmapakket van een logische simulator. Met deze programmatuur is het mogelijk om digitale schakelingen te simuleren, door hier een computermodel van te maken en daar de simulator op los te laten. De simulator rekent het model dan gedurende een van te voren opgegeven tijdsinterval door, waarna de te onderzoeken signalen bekeken kunnen worden.
Bij de simulator is uitgegaan van zowel flexibiliteit als nauwkeurigheid. De flexibiliteit is verwezenlijkt door de simulator te voorzien van een "bibliotheek" met standaardcomponenten. In deze bibliotheek zijn de elementaire bouwstenen, zoals AND, OR, EXOR etc, opgeslagen en deze bibliotheek kan door de gebruiker met eigen componenten uitgebreid worden (hiervoor is wel enige kennis van Boolse algebra noodzakelijk). Deze constructie biedt ook de mogelijkheid om reeds getestte en goed bevonden schakelingen als standaardcomponent op te nemen.
Ter bevordering van de nauwkeurigheid is in de simulator de mogelijkheid ingebouwd om rekening te houden met de delays van bouwstenen. In de praktijk komt dit er op neer dat geen bouwstenen met delay 0 voorkomen. De gebruiker kan bij elke bouwsteen die hij gebruikt de gewenste delaywaarde opgeven. Bij een niet gespecificeerde waarde wordt door de simulator de defaultwaarde 1 ingevuld. Deze "1" is de door de simulator gebruikte tijdseenheid. Men kan hier zelf een betekenis aan toe kennen in de vorm van seconde, microseconde etc.
De simulator werkt alleen met de logische waarden 0 (laag niveau) en 1 (hoog niveau). Doordat er niet met spanningen wordt gewerkt zijn alle signalen dus rechthoekvormig. Bij toepassing van spanningen zou het programma te gecompliceerd en te lang worden en de rekentijd zou te groot worden. Hierdoor houdt de simulator geen rekening met te grote belastingsweerstanden (te veel bouwstenen aangesloten op een uitgang) of belastingscapaciteiten, waardoor de signalen bij grotere frequenties vervormen. Om dit te controleren is men dan ook aangewezen op de datasheets van de toegepaste IC's.
Na deze eerste kennismaking volgt nu een globale beschrijving van de simulator. Daarna zal het programma-pakket zelf besproken worden en de toepassing hiervan.

2. Benodigheden

De simulator gebruikt het geheugengebied #8200 t/m #9FFF en het gebied #2900 t/m 3FFF. Het geheugen vanaf #2900 wordt gebruikt voor opslag van een programma en opslag van het computermodel, tabellen, lijsten en tussenresultaten. Hoe groter dit geheugengebied, hoe groter de te simuleren schakeling kan zijn.
Als het programma loopt mogen er geen andere programma's of boxen, die gebruik maken van de zeropage van #80 t/m #99, actief zijn. Verder wordt er ook gebruik gemaakt van de zeropage van #60 t/m #7B.

Naast al deze geheugenbenodigheden moet er ook een disk drive aanwezig zijn. De simulator slaat namelijk de resultaten op in een disk file.

3. Toegepast principe

Voor het construeren van digitale simulators zijn twee basisprincipes toepasbaar. Het eerste principe wordt de "compiler driven simulator" genoemd. Hierbij wordt het model van de schakeling omgezet naar "machine executable code". In het geval van de Acorn Atom is dit de machinetaal van de 6502. Voor de simulatie van een 3-input AND-gate met ingangen B,C,D en uitgang E krijgt men dan:

```
LDA B;AND C;AND D;STA E
```

De simulator voert nu voor elke inputvector de "compiled code" uit en bepaald de uitgangswaarden. Om bij dit principe gebruik te kunnen maken van delays moet een aparte bouwsteen, de delay, ingevoerd worden.
Het tweede principe, dat ook hier is toegepast, is de "event directed simulation". Hierbij wordt, in tegenstelling tot de eerste methode, niet iedere keer de gehele schakeling doorgerekend, maar alleen die gedeelten waar signaalveranderingen optreden. Stel op tijdstip t verandert een ingang van bouwsteen x met delay y. De simulator rekent bouwsteen x dan geheel door en als de nieuwe uitgangswaarde verschilt van de oude, dan zal op tijdstip t+y dus een signaalverandering optreden. Deze informatie wordt dan opgeslagen in een tijdlijst. In deze tijdlijst worden alle veranderingen in chronologische volgorde opgeslagen. Als de simulator bij het doorlopen van de tijdlijst nu is aangekomen bij tijdstip t+y, dan worden alle bouwstenen die op bouwsteen x zijn aangesloten gesimuleerd. Als nu bijvoorbeeld een van die bouwstenen w is met delay z en hier treedt ook weer een signaalverandering op aan de uitgang, dan wordt dit opgeslagen in de tijdlijst op tijdstip t+y+z. Treedt er geen verandering op dan gebeurt er niets.
De simulator is dus constant bezig met het halen van informatie uit de tijdlijst, het onderzoeken van mogelijke veranderingen en het weer terugstoppen van informatie in de tijdlijst. Dit gaat net zolang door totdat het te onderzoeken interval geheel is onderzocht of er geen veranderingen meer optreden (schakeling is stabiel).
Het tweede principe biedt enkele voordelen t.o.v. het eerste principe. Ten eerste is de methode veel efficienter omdat hij alleen rekent als er iets in de schakeling gebeurt. Vervolgens wordt er rekening gehouden met vertragingen in de bouwstenen. En tenslotte kan bij dit principe elk soort schakeling bekeken worden, bij het eerste principe kunnen alleen bouwstenen bekeken worden, waarvoor code gegenereerd kan worden.

4. Toegepaste logica

Bij de simulator wordt gebruik gemaakt van een beperkte vorm van de vijf waarden logica. Naast de normale logische waarden "0" en

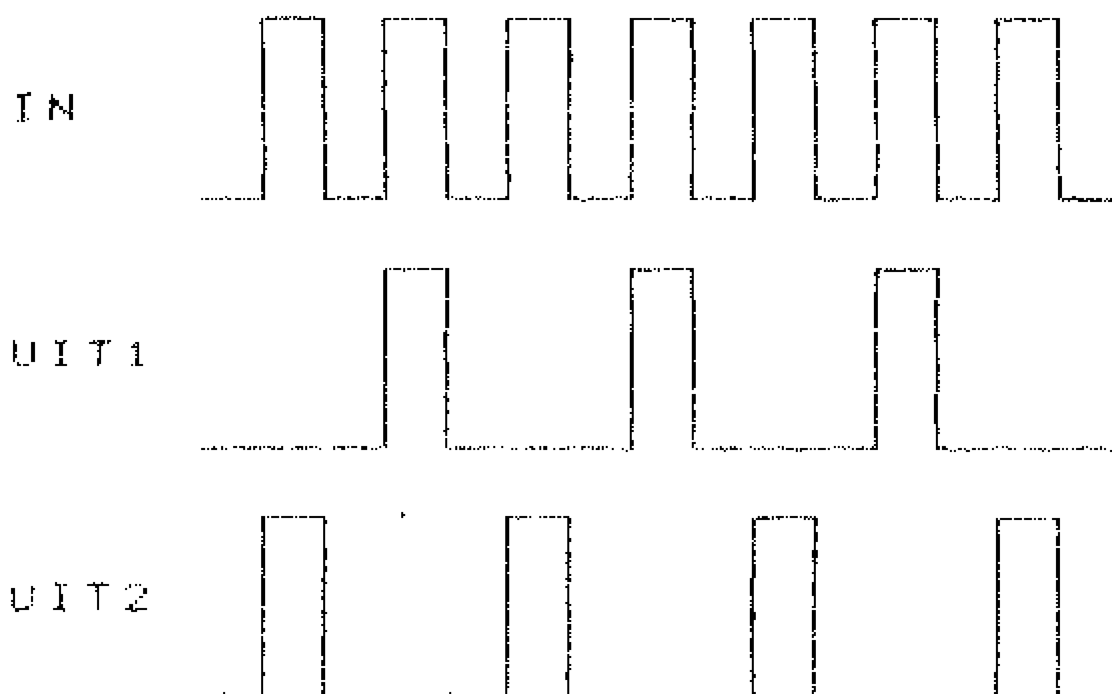
"1" bevat deze logica ook de waarden "u" (unknown), Ø/1- en 1/Ø-overgang. De waarden "Ø" en "1" worden normaal gebruikt als ingangswaarden en uitgangswaarden van de bouwstenen. De waarde "u" wordt als beginwaarde gebruikt. Bij het begin van de simulatie worden alle signalen op de waarde "u" gezet. Zo kan onderzocht worden hoelang het duurt voordat een signaal een gedefinieerde waarde aanneemt en of er misschien signalen zijn die niet beinvloedbaar zijn. Bij het rekenen met "u" gelden de volgende regels (& stelt AND voor,¦ OR):

u & Ø = Ø
u & 1 = u
u ¦ Ø = u
u ¦ 1 = 1

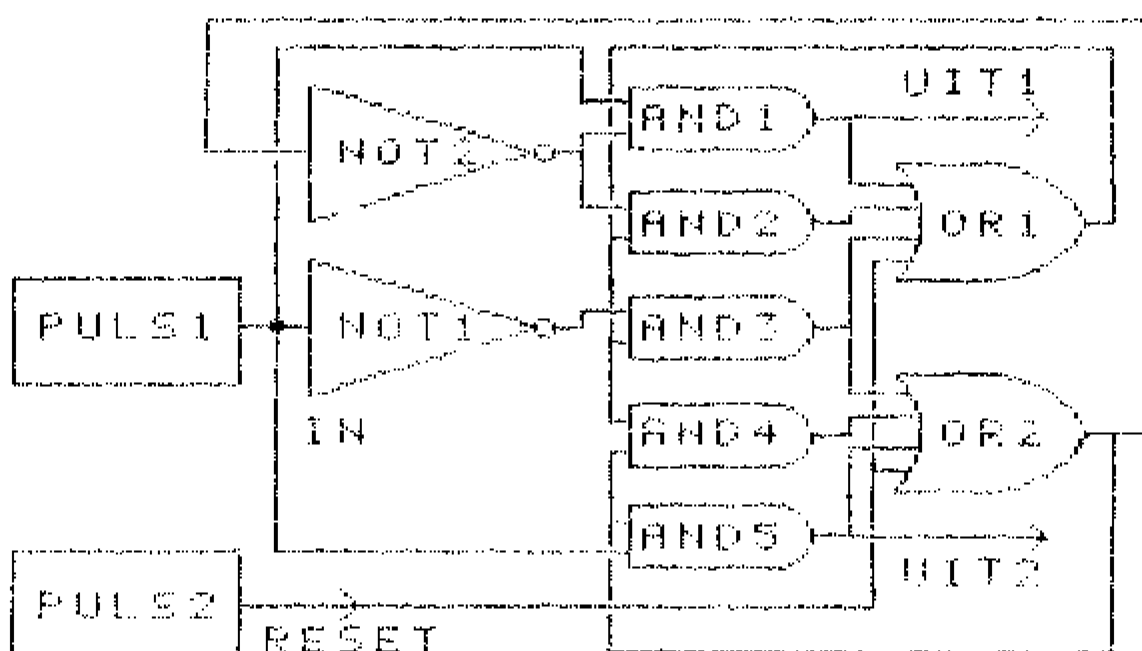
Bij de output van de simulatie wordt "u" weergegeven als een niveau dat halverwege het Ø- en 1-niveau ligt.
De waarden Ø/1 en 1/Ø worden alleen gebruikt bij het al eerder genoemde doorrekenen van de bouwstenen tijdens het onderzoeken van een mogelijke verandering. Bij alle ingangen wordt dan, voordat aan het doorrekenen begonnen wordt, d.m.v. een bit aangegeven of het een overgang is of niet. Hierdoor is het mogelijk om bouwstenen te ontwerpen die alleen op overgangen reageren (bijv. positive edge-triggered D-flipflop).

5. De programmatuur

Na de blik in het inwendige van de simulator is dan nu het programmapakket zelf aan de beurt. De programmatuur bestaat uit 2 programma's. Het eerste programma, LOSIM-Ø, is de eigenlijke simulator. Dit programma heeft als input het computermodel van de schakeling nodig en genereert als output een file, waarin de te onderzoeken signalen zijn opgeslagen. Verder kan men met dit programma nieuwe standaardcomponenten in de bibliotheek opslaan. LOSIM-Ø wordt in het geheugen geladen vanaf #8200, terwijl het model geladen wordt vanaf #2900. LOSIM-Ø bestaat uit een BASIC-gedeelte, dat 1K beslaat, gevolgt door een machinetaalgedeelte van 4 3/4K. Hierdoor blijft 1 3/4K over voor de bibliotheek.
Met het tweede programma, DISPLAY, kunnen de signalen bekeken worden. Het programma heeft als input de outputfile van LOSIM-Ø en genereert hieruit een plaatje op het scherm in graphics mode 4. DISPLAY wordt in het geheugen geladen vanaf #2900.
De verschillende aspecten van de simulator zullen nu besproken worden. Dit zal toegelicht worden aan de hand van een voorbeeld. Bij dit voorbeeld gaat het om een schakeling die een blokgolf omzet in een niet overlappende tweefase klok. In figuur 1 staat bovenaan het ingangssignaal (de blokgolf) en daaronder staan de 2 uitgangssignalen van de schakeling. M.b.v. een speciale ontwerptechniek kan hiervoor een asynchrone schakeling geconstrueerd worden, die is weergegeven in figuur 2. Hierin is PULS1 het ingangssignaal en PULS2 een resetsignaal. M.b.v. de simulator kan nu getest worden of de ontworpen schakeling ook voldoet.



Figuur 1



Figuur 2

5. Het model

Alvorens men kan simuleren moeten ten eerste de gewenste standaardcomponenten in de bibliotheek zijn opgeslagen. Hierover straks meer. Verder moet het model van de digitale schakeling aanwezig zijn. Dit model beschrijft de complete schakeling. Het wordt opgemaakt net zoals een BASIC-programma, d.w.z. m.b.v. regels met regelnummers.

Om te beginnen moeten namen worden toegekend aan de verschillende bouwstenen en verbindingsdraden. In figuur 2 is dit al gedaan voor de bouwstenen. Naast de eigenlijke schakeling wordt dit ook gedaan voor de verschillende spanningsbronnen die op de schakeling zijn aangesloten. De simulator beschikt over 2 soorten spanningsbronnen. De eerste is de SET, een gelijkspanningsbron, waarmee op een lijn een lage of hoge spanning gezet kan worden ("Ø" of "1"). De tweede is de PULS, een generator voor symmetrische of asymmetrische blokgolven (zie PULS1 en PULS2 figuur 2).

Als dit voltooid is kunnen de drie blokken van het model opgesteld worden. Het eerste blok is het WIRE-blok. Dit blok begint met "WIRE", waarna alle bedradingen opgesomd moeten worden, onderling gescheiden door komma's en afgesloten met ";". In figuur 3, het totale model, is dit ook terug te vinden.
Het tweede blok is het COMPOUND-blok. Dit blok begint met "COMPOUND", waarna alle bouwstenen vermeld moeten worden met vermelding van het type (welke standaardcomponent). Bouwstenen die van hetzelfde type zijn kunnen worden opgenomen als een rij, afzonderlijk gescheiden door komma's en afgesloten met ":". Hierna moet de standaardcomponent vermeld worden, waarna alles wordt afgesloten met ";". De constructie is dus als volgt:

```
COMPOUND1,COMPOUND2,...,COMPOUNDx      :COMPONENTy;
```

Figuur 3 kan weer geraadpleegt worden voor de verduidelijking. Het is niet noodzakelijk om alle bouwstenen van hetzelfde type in een rij op te nemen. Een rij kan ook opgesplitst worden in kleinere sub-rijen van het zelfde type. Zo kan regel 50 uit figuur 3 vervangen worden door:

```
50 AND1,AND2,AND3       :AND2;
55 AND4,AND5            :AND2;
```

Als een rij meerdere regels in beslag neemt moet er wel voor gezorgd worden dat een naam (dit geldt trouwens voor het hele model) op een en dezelfde regel staat en niet dat bijv. de eerste 3 letters op het einde van een regel staan en de rest aan het begin van de volgende regel. Verder is het verboden om in een naam spaties op te nemen (geldt ook voor het hele model).
Na deze eerste twee blokken komt het derde en laatste blok. Dit begint met "BEGIN" en eindigt met "END". In dit blok moet worden aangegeven welke draden op de bouwstenen zijn aangesloten. Dit moet voor elke bouwsteen apart gedaan worden. Eerst wordt de bouwsteen-naam vermeld, gevolgd door "(". Hierna volgen, gescheiden door komma's, de ingangsdraden en vervolgens de uitgangsdraden. De volgorde hiervan is vastgelegd in de standaardcomponenten. Bij een AND of OR maakt het niet uit in welke volgorde men de ingangen plaatst, omdat de ingangen hier gelijkwaardig zijn. Bij bijvoorbeeld een flipflop is de volgorde wel van belang.
Als nu alles wordt afgesloten met ");", dan heeft men een bouwsteen met delay 1. De constructie is dus als volgt:

```
COMPOUNDx(IN1,IN2,...,INy,UIT1,...,UITz);
```

Zie ook weer figuur 3. Als een andere delay waarde gewenst is, kan die opgenomen worden voor ");". Stel AND4 heeft een delay 12, dan wordt regel 190:

```
190 AND4(OR3OUT,OR4OUT,AND4OUT,12);
```

Een uitzondering op dit verhaal vormen de SET-bouwsteen en de PULS-bouwsteen. De constructie voor SET is:

```
SETx(WIREy,z);  met z=0 of z=1
```

SET5(IN3,0); wil zeggen dat op IN3 spanningsbron SET5 wordt aangesloten met een lage spanning (niveau "0"). De constructie

voor PULS is:

PULSx(WIREy,a,b,c); met de gehele getallen a>=0, b>0 en c>0

PULS5(IN2,5,100,30); wil zeggen dat op IN2 pulsgenerator PULS5 wordt aangesloten met periodetijd 100, een pulshoogtijd van 30 en de eerste puls komt op tijdstip 5.
Als op bovenstaande manier een model is gemaakt, kan dit gesaved en weer geladen worden als een normaal BASIC-programma.
WAARSCHUWING: alle namen uit het WIRE-blok en het COMPOUND-blok moeten in het BEGIN-END blok voorkomen!!

```
 10WIRE
 20 IN,RESET,NOT1OUT,NOT2OUT,AND1OUT,AND2OUT,AND3OUT,
 30 AND4OUT,AND5OUT,OR1OUT,OR2OUT;
 40COMPOUND
 50 AND1,AND2,AND3,AND4,AND5       :AND2;
 60 OR1,OR2                        :OR4;
 70 NOT1,NOT2                      :NOT;
 80 PULS1,PULS2                    :PULS;
 90BEGIN
100 PULS1(IN,10,100,50);
110 PULS2(RESET,2,1000,5);
120 NOT1(IN,NOT1OUT);
130 NOT2(OR2OUT,NOT2OUT);
140 OR1(RESET,AND1OUT,AND2OUT,AND3OUT,OR1OUT);
150 OR2(RESET,AND3OUT,AND4OUT,AND5OUT,OR2OUT);
160 AND1(IN,NOT2OUT,AND1OUT);
170 AND2(NOT2OUT,OR1OUT,AND2OUT);
180 AND3(NOT1OUT,OR1OUT,AND3OUT);
190 AND4(OR1OUT,OR2OUT,AND4OUT);
200 AND5(IN,OR2OUT,AND5OUT);
210END
```

Figuur 3

6. LOSIM-0

Als het model in het geheugen staat vanaf #2900 kan de simulator gestart worden met *LOAD"LOSIM-0", ?18=#82, OLD en RUN. Als eerste moet de laatste vrije geheugenplaats vanaf #2900 ingevoerd worden. Als het geheugen is uitgebreid met de 16K RAM kaart voert men hier dus #7FFF in.
LET OP: het gaat om de laatste aanwezige geheugenplaats, niet de eerste afwezige!!
Na deze systeeminformatie volgt een menu waaruit een keuze gemaakt kan worden. Met keuze 3 en 4 wordt een overzicht verkregen van de aanwezige standaardcomponenten met aanvullende informatie (beginadres) op het scherm resp. de printer. Bij dit overzicht zal men PULS en SET niet aantreffen, omdat deze in de simulator zijn ingebouwd. Met keuze 5 wordt het programma verlaten en is men terug in textspace #2900. Tevens wordt een OLD uitgevoerd. Als er een fout in het model is geconstateerd kan men na keuze 5 meteen de fout opzoeken en verbeteren. Na ?18=#82, OLD en RUN kan er weer gesimuleerd worden. Verder wordt de laatst bezette geheugenplaats van het programma op het scherm afgedrukt. Als er nieuwe standaardcomponenten aan de bibliotheek zijn

toegevoegd, moet het programma opnieuw gesaved worden. Stel er wordt 9F13 afgedrukt, dan moet het programma gesaved worden met *SAVE"LOSIM0x"8200 9F14.
LET OP: het gaat om de laatst bezette geheugenplaats, dus saven met 1 meer!!
Er wordt aangeraden eigen versies op te slaan volgens de nummering LOSIM0x. Eventuele verbeterde versies van LOSIM-0 kunnen dan genummerd worden volgens LOSIM-1, LOSIM-2 etc.
De eerste twee keuzes zullen nu uitvoerig besproken worden.

6.1 Opslaan van standaardcomponenten

Na keuze 2 krijgt men toegang tot de bibliotheek. Er kan nu een nieuwe standaardcomponent ingevoerd worden. Als er bij het invoeren iets mis gaat, wordt naar het menu teruggekeerd en wordt de tot dan toe gedeeltelijk ingevoerde standaardcomponent uitgewist. Er moet dan weer van voren af aan begonnen worden. Een foutmelding kan op twee manieren tot stand komen. Ten eerste door een fout bij de invoer. Er kan dan gewoon opnieuw begonnen worden met de invoer. Ten tweede kan het voorkomen dat de bibliotheek vol is, d.w.z. door de nieuwe standaardcomponent zou het totale programma #9FFF overschrijden.
Bij de invoer moet als eerste de naam, het aantal ingangen en het aantal uitgangen van de standaardcomponent ingevoerd worden. Vervolgens moet voor elke uitgang de functie Fx, die de samenhang tussen uit- en ingangen weergeeft volgens UITx=Fx(IN1,IN2,...,INy), ingevoerd worden.
Bij de invoer moeten de ingangen aangeduid worden met hoofdletters, beginnende vanaf A. Als er bijvoorbeeld 3 ingangen zijn dan komt A overeen met ingang 1, B met 2 en C met 3. Een andere dan deze drie letters resulteert in een foutmelding. Er kan ook aangegeven worden dat het om de inverse van de ingang gaat door een kleine letter te gebruiken. Als laatste bestaat de mogelijkheid om de ingang alleen op een flank te laten reageren door achter de letter een "/" te plaatsen. Ze geeft "A/" aan dat er gereageerd wordt op een stijgende flank op ingang 1 (overgang van 0 naar 1) en "a/" geeft aan dat het om een dalende flank gaat (overgang van 1 naar 0).
Dan zijn we tenslotte bij de functies zelf aan gekomen. Deze moeten samengesteld worden volgens de "sum-of-products" vorm oftewel de "minterm" vorm. De notatie hiervoor is:

Fx=ABC..Z+aBC..Z+.....

Dit verklaart de benaming, som van produkten. Som stelt hier de OR-functie voor en de produkten zijn AND-functies. De simulator kent dus alleen de logische functies AND en OR en elke functie moet nu volgens de opgegeven notatie in die twee basisfuncties uitgedrukt worden. Ter verduidelijking volgen nu enkele voorbeelden van verschillende functies:

inverter (7404) :	a
2-input AND gate (7408) :	AB
2-input OR gate (7432) :	A+B
4-input NAND gate (7420) :	a+b+c+d
3-input NOR gate (7427) :	abc
2-input EXOR gate (7486) :	aB+Ab

2-input EXNOR gate (74266) : AB+ab
positive edge-triggered D-flipflop with clear (74174) : AB/C
met A=D-ingang, B=CLOCK en C=non-CLEAR
JK flipflop with clear (7473) : abCd/E+Abd/E+ABcd/E
met A=J, B=K, C=Q, D=CLOCK en E=non-CLEAR

Voor het gebruik van "/" is nog enige toelichting nodig. De toepassing van flankgevoelige ingangen is vooral ingebouwd voor gebruik in flipflops. Als in een produktterm een flankgevoelige ingang voorkomt en er staat op die ingang geen flank maar een constant niveau, dan worden alle voorgaande ingangen in die produktterm vervangen door de huidige uitgangswaarde. Men moet bij de invoer van de standaardcomponent hier dus rekening mee houden. Zo kan bij de D-flipflop de functie, in geval van een stijgende flank op B vervangen worden door A.1.C=AC, en zonder flank door QC (Q=uitgang). Bij de JK-flipflop wordt de functie bij een dalende clock flank abCE+AbE+ABcE (=(jkQ+Jk+JKq).clear) en bij constant clock niveau QE+QE+QE=QE (=Q.CLEAR). Bij het laatste voorbeeld valt op dat de uitgang ook als ingang voorkomt. Dit zal vaak het geval zijn bij geheugenelementen. Bij de specificatie van een bouwsteen krijg je dan de volgende constructie:

JK-FLIPFLOP(J,K,Q,CLOCK,CLEAR,Q);

Laten we als laatste punt nog even kijken naar het voorbeeld van de tweefase klok. Voordat het model gesimuleerd kan worden, moeten de volgende standaardcomponenten ingevoerd worden:

naam :	AND2	OR4	NOT
aantal ingangen :	2	4	1
aantal uitgangen :	1	1	1
functie :	AB	A+B+C+D	a

6.2 Simuleren

Na keuze 1 kan de schakeling gesimuleerd worden. Eerst wordt het model door her subprogramma PREPARE gecontroleerd op fouten en worden de tabellen en lijsten bepaald die nodig zijn voor de eigenlijke simulatie. Bij het voorbeeld van de tweefase klok neemt dit ongeveer 2 seconde in beslag. Als er bij het controleren van het model een fout wordt geconstateerd, zal de mededeling "ERROR AT LINE xxx" verschijnen met daaronder de aard van de fout. De volgende foutmeldingen kunnen voorkomen:

PROGRAM INCOMPLETE - er is geen model aanwezig
- niet alle drie de blokken zijn volledig
- er zijn geen spanningsbronnen gespecificeerd

OVERFLOW - de getallen bij een PULS-bouwsteen zijn groter dan 2^31-1

UNKNOWN WIRE - een wire uit het BEGIN-END blok komt niet voor in het WIRE-blok

UNKNOWN COMPOUND - een bouwsteen uit het BEGIN-END blok komt niet voor in het COMPOUND-blok

UNKNOWN COMPONENT - een component uit het COMPOUND-blok komt

	niet voor in de bibliotheek
MEMORY FULL	- er is niet genoeg geheugen meer om alle informatie op te slaan (zal zelden of nooit voorkomen)
TO MANY COMPOUNDS	- er zijn meer dan 255 bouwstenen (erg onwaarschijnlijk)
TO MANY WIRES	- er zijn meer dan 255 wires (erg onwaarschijnlijk)
SYNTAX	- er zit een typfout in een van de woorden WIRE, COMPOUND, BEGIN en END
	- in een PULS-bouwsteen is periodetijd < pulshoogtijd
	- er komt een compound- of wire-naam voor met lengte 0 (2 komma's, komma en ":" of komma en ";" achter elkaar)
	- in een SET-bouwsteen wordt een andere waarde dan 0 of 1 opgegeven
	- ontbreken van ")" of ";"
	- de opgegeven getallen zijn geen geldige decimale getallen
	- het aantal ingangen en uitgangen van een bouwsteen komt niet overeen met de standaardcomponent

Na al deze foutmeldingen komt men met een druk op een toets terug bij het menu.
Na deze voorbereiding moet opgegeven worden hoeveel en welke wires na de simulatie bekeken moeten worden. Hier moeten namen uit het WIRE-blok opgegeven worden. Gebeurt dit niet dan komt de foutmelding:

```
ERROR AT LINE yyy
UNKNOWN WIRE
```

waarbij yyy het regelnummer is waarin "END" staat. Ook nu komt men weer terug bij het menu. Bij ons voorbeeld moeten we achtereenvolgens 4, IN, RESET, AND1OUT en AND5OUT invoeren.
Vervolgens moet nog opgegeven worden hoe lang er gesimuleerd moet worden. Omdat bij het voorbeeld generator PULS1 een periodetijd heeft van 100 is 500 een geschikte waarde.
Als laatste wordt naar de naam van de outputfile gevraagd, waarna door het subprogramma SIMUL de eigenlijke simulatie uitgevoerd kan worden. Dit duurt bij het voorbeeld ongeveer 3 seconde. Ook nu kunnen er weer fouten optreden, te weten:

MEMORY FULL	- het geheugen is te klein om de simulatie voort te kunnen zetten
TIME OVERFLOW	- er is een tijdstip > $2^{31}-1$ bereikt

De laatste twee foutmeldingen hoeven nog niet te betekenen dat de simulatie voor niets is geweest. De outputfile en het programma "DISPLAY" zijn zo algemeen van opzet dat het altijd mogelijk is om te bekijken wat er nog wel door de simulator is uitgerekend.

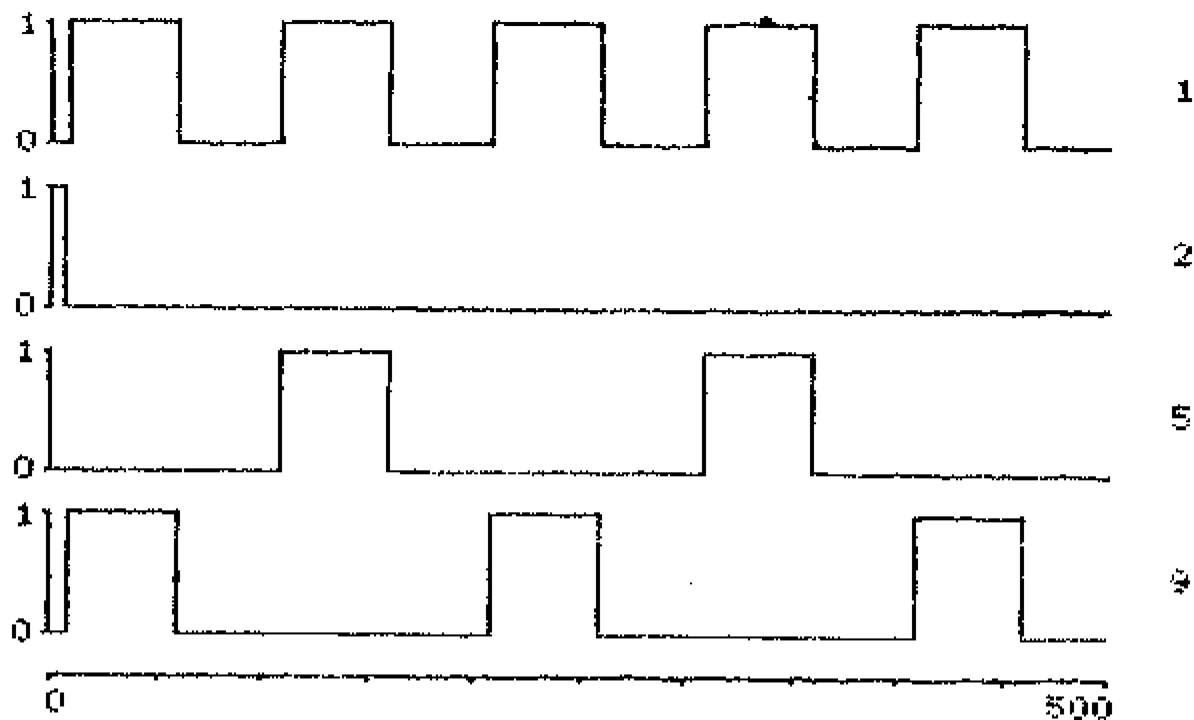
7. DISPLAY

Nadat de simulatie voltooid is, kan men LOSIM-0 verlaten en DISPLAY in het geheugen laden. Wil men LOSIM-0 of het simulatiemodel nog opslaan, dan moet dit eerst gedaan worden. Het resultaat van de simulatie uit het voorbeeld is te vinden in figuur 4.

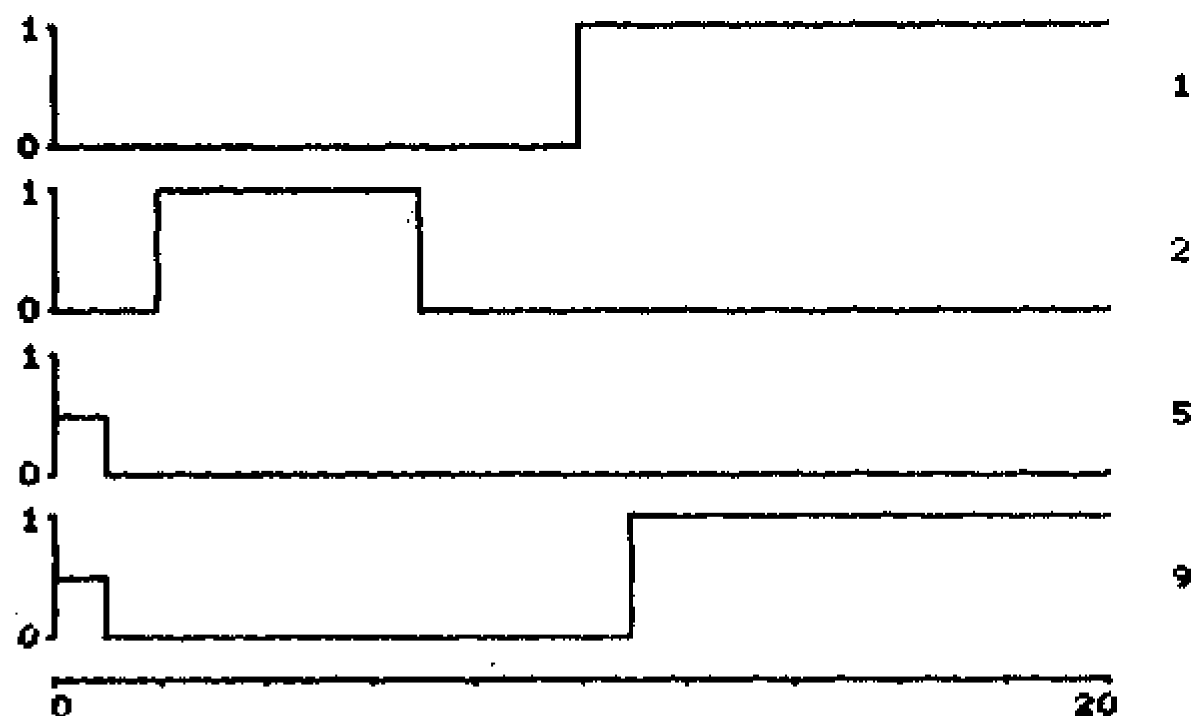
Met DISPLAY kan men elk gewenst interval bekijken. Als het programma voor de eerste keer doorlopen wordt kan het beste het interval bekeken worden dat door de simulator is uitgerekend (0 tot opgegeven waarde). Nadat de afbeelding klaar is kan dan na een druk op een toets een nieuw interval bekeken worden, zodat bepaalde delen kunnen worden uitvergroot (zie figuur 5). Als een interval wordt opgegeven dat groter is dan het berekende interval, worden de signalen na de laatste berekende waarde tot het einde doorgetrokken.

Na de afbeelding kan het plaatje ook uitgeprint worden. Met "O" wordt een klein plaatje verkregen en met "P" een grote afdruk. De ingebouwde routines zijn bestemd voor een SEIKOSHA GP-250X. Als men een andere printer heeft, moet het assemblergedeelte van regel 110 t/m 690 en ook regel 1230 en 1240 vervangen worden (array's AA, CC, DD, EE, FF, MM, XX en YY niet gebruiken).

Bij elk signaal wordt rechts een WIRE-nummer afgedrukt. In het WIRE-blok is terug te vinden bij welke wire het nummer hoort, door vanaf de eerste wire te tellen, beginnende met 1. Zo komt in figuur 4 1 overeen met IN, 2 met RESET, 5 met AND1OUT en 9 met AND5OUT (zie figuur 3).



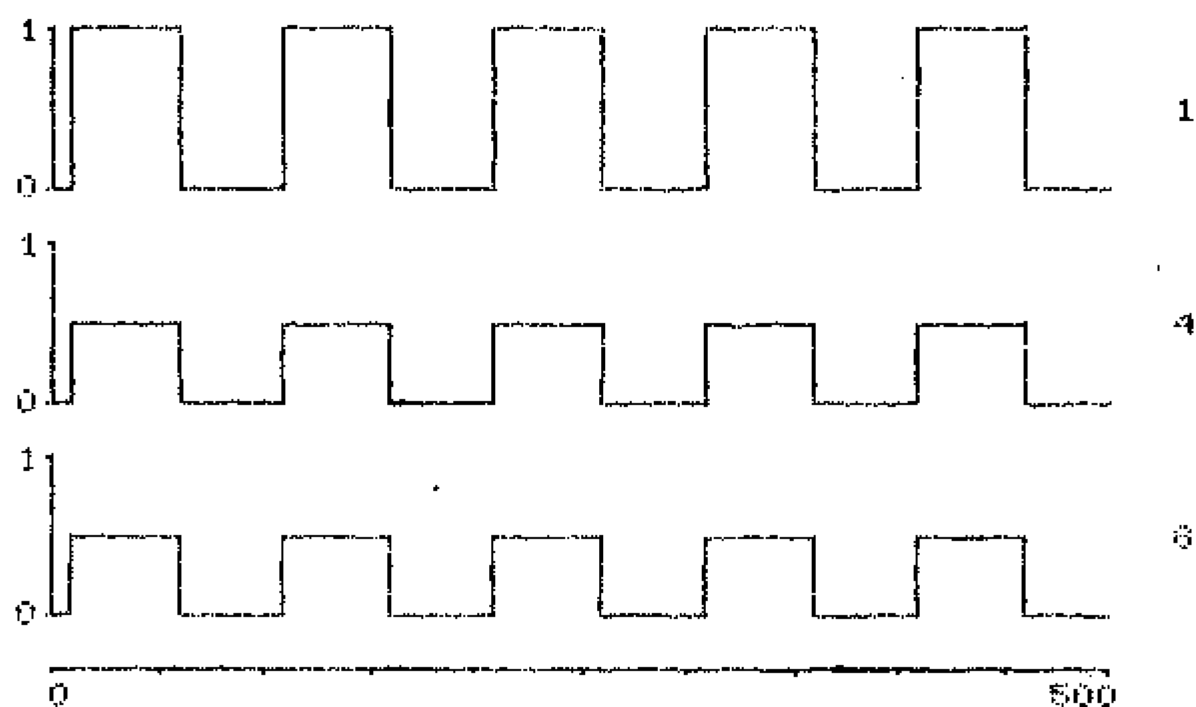
Figuur 4



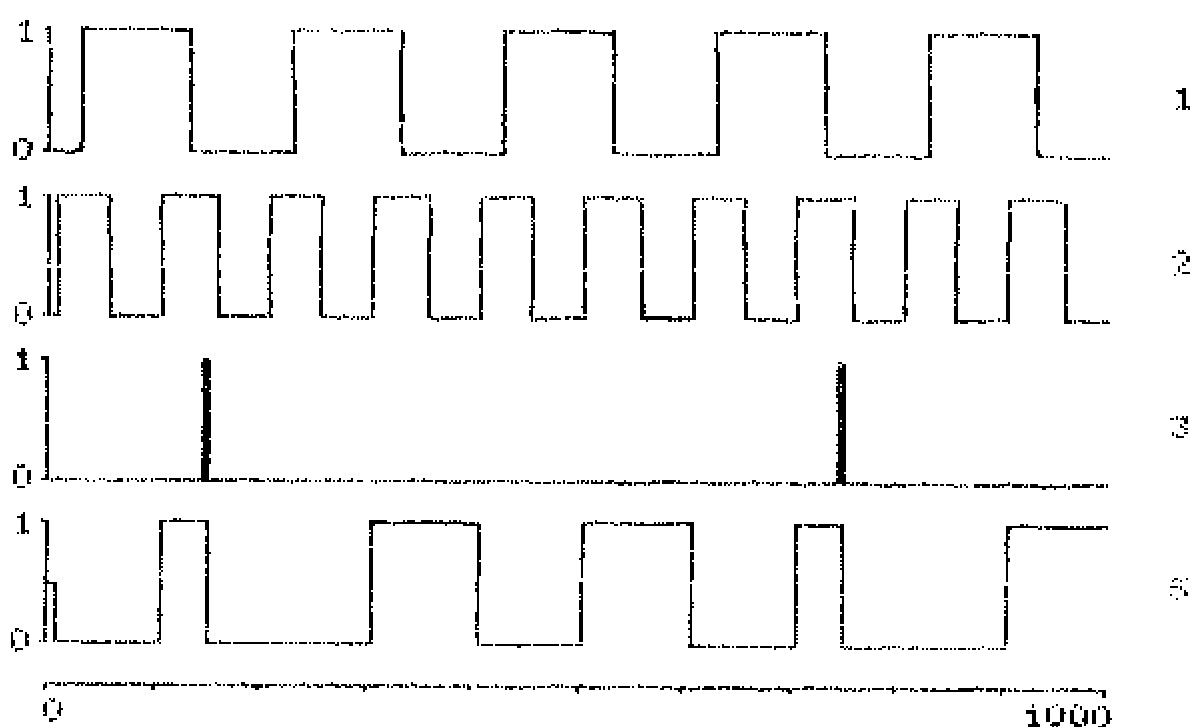
Figuur 5

8. Nog enkele voorbeelden

In figuur 6 is te zien wat er gebeurt als in het voorbeeld de reset wordt weggelaten. Doordat er niet gereset wordt, zal de simulator bij alle uitgangen de waarde "u" invullen. Alleen NOT1 en een ingang van AND4 en AND5 hebben nu een gedefinieerde waarde. Volgens de simulatie zou de schakeling het nu dus niet doen. In de praktijk zal dit wel meevallen, omdat elke bouwsteen wel een bepaalde waarde zal aannemen en de schakeling dus toch in een bepaalde begintoestand zal komen. De simulator is dus erg pessimistisch. Je kunt het ook als volgt uitdrukken: als de simulator een schakeling goedkeurt weet je ook zeker dat hij het doet.
In figuur 7 is een simulatie van de al eerder vermeldde D-flipflop weergegeven. Van boven naar beneden zijn achtereenvolgens D, CLOCK, CLEAR en Q uitgezet. Uit deze figuur blijkt duidelijk de toepassing van "/". Op elke stijgende clock-flank wordt de informatie op de D-ingang overgenomen en in alle andere gevallen blijft de uitgang hetzelfde (behalve bij een reset).



Figuur 6



Figuur 7

9. Nawoord

Aanzet tot dit artikel en de programma's was de overweging of het mogelijk is om op een microcomputer een relatief snelle simulator te laten draaien. Ik geloof dat ik dat met LOSIM-Ø wel heb bewezen, alhoewel ik tot nu toe alleen nog maar kleine schakelingen heb uitgeprobeerd. Hoe het programma zich gedraagt bij grotere schakelingen moet nog blijken. Hierbij maak ik me niet zo zeer zorgen om de benodigde geheugenruimte, omdat bij alle voorbeelden is gebleken dat het programma gemakkelijk kan draaien zonder de aanwezigheid van een 16K ram kaart. Als een 16K ram kaart aanwezig is hoeft men zich dus nergens zorgen om te maken. Het enigste probleem bij grote schakelingen kan de rekentijd opleveren. Mocht dit grote problemen opleveren, dan zal geprobeerd moeten worden om een snellere opvolger van LOSIM-Ø te maken.

Een ander uitgangspunt voor een opvolger zou kunnen zijn om een

versie te maken die geen disk drive nodig heeft. Ik heb het programma niet met een cassette recorder uitgeprobeerd, maar ik verwacht niet dat dat zal lukken. Het opslaan van de data in de file wordt namelijk door een machinetaal programma gedaan, terwijl het halen van de data uit de file door een basic programma gebeurt (DISPLAY).
De programma's zelf zijn niet opgenomen in dit artikel. Dit omdat het belangrijkste deel bestaat uit machinetaal met een source van ruim 19K en het is ondoenlijk dit zonder fouten in te typen.
Naast de programma's LOSIM-0 en DISPLAY is nog een derde programma beschikbaar, namelijk LOSIM01, die in tegenstelling tot LOSIM-0 geen lege bibliotheek heeft. De volgende standaardcomponenten zijn aanwezig:

omschrijving	definitie
inverter	NOT(IN,UIT);
2-input and gate	AND2(IN1,IN2,UIT);
3-input and gate	AND3(IN1,IN2,IN3,UIT);
4-input and gate	AND4(IN1,IN2,IN3,IN4,UIT);
2-input nand gate	AND2(IN1,IN2,UIT);
3-input nand gate	NAND3(IN1,IN2,IN3,UIT);
4-input nand gate	NAND4(IN1,IN2,IN3,IN4,UIT);
2-input or gate	OR2(IN1,IN2,UIT);
3-input or gate	OR3(IN1,IN2,IN3,UIT);
4-input or gate	OR4(IN1,IN2,IN3,IN4,UIT);
2-input nor gate	NOR2(IN1,IN2,UIT);
3-input nor gate	NOR3(IN1,IN2,IN3,UIT);
4-input nor gate	NOR4(IN1,IN2,IN3,IN4,UIT);
exor gate	EXOR(IN1,IN2,UIT);
exnor gate	EXNOR(IN1,IN2,UIT);
D-flipflop (74174)	D-FLIPFLOP(D,CLOCK,NON-CLEAR,Q);
JK-flipflop (7473)	JK-FLIPFLOP(J,K,Q,CLOCK,CLEAR,Q);

OPROEP

Wie beschikt over de software voor de ATOM om hiermee print-lay-outs te maken/ontwerpen.
Stuur je reaktie s.v.p. naar de redaktie.

INHOUDSOPGAVE:

TEKSTVERWERKER ACORN-ATOM

INLEIDING.

Met de tekstverwerker kunnen documenten zoals brieven, folders of boeken gemaakt worden. De tekst kan achter elkaar door ingevoerd worden en hoeft niet in zinnen afgebroken te worden. Dit doet de Editor automatisch.
In de Atom met geheugen van #2800 tot #3C00 kunnen ongeveer 5000 tekens bewerkt worden, wat overeenkomt met ongeveer 3 pagina's A4 tekst. Er is ook een zgn. correctiebuffer, een geheugengedeelte waarin tijdelijk een stuk tekst kan worden opgeslagen om het later te gebruiken. (maximaal 448 tekens).

AANZETTEN EDITOR:

Het aanzetten van de editor gebeurt door EDIT te tikken en daarna op de toets RETURN te drukken.
Als er geen floating point ROM in de Atom aanwezig is, moet de editor aangezet worden met LINK #ACFD, RETURN.
Op het scherm verschijnt de vraag "Old Text?".
Als er geen tekst in het geheugen zit, die bewerkt moet worden, kan er op RETURN gedrukt worden. Daarna wordt het scherm leeggemaakt en verschijnt er een wit blokje dat het "einde tekst" aangeeft. Op de onderste regel verschijnt het nummer van de "beeldscherm-pagina met daarachter de zgn. prompt die aangeeft dat de editor voor gebruik gereed is. Dit ziet er als volgt uit:

p01

INTIKKEN TEKST:

Na intikken van het commando E verschijnt op de onderste regel achter de prompt E:
Het geheel gaat er als volgt uitzien:

p01>E:

Nu kan er tekst ingetikt worden. Dit kan achter elkaar door gebeuren en de tekst hoeft niet in zinnen afgebroken te worden. Dit doet de Editor automatisch. Een nieuwe regel kan begonnen worden met de toets RETURN. Een regel overslaan wordt gedaan door een lege regel te maken (2x RETURN).
Als een woord niet op de regel past hoeft het NIET te worden afgebroken, maar kan het op de volgende regel doorgetikt worden.
Het intikken van tekst kan beeindigd worden door op de toets COPY te drukken. Hierna wordt de laatste schermpagina tekst getoond

BEKIJKEN INGETIKTE TEKST:

Voor het bekijken van tekst zijn de volgende commando's beschikbaar:
S (start of text.) Dit toont de eerste schermpagina vanaf het begin van de tekst.
Z (end of text) Toont de laatste schermpagina tot het einde

van de tekst.

N (next page) Toont een volgende schermpagina.
P (previous page) Toont de voorafgaande schermpagina.
[(roll up a line) Schuift de tekst 1 regel omhoog.

CURSORBESTURING:

De op het scherm getoonde tekst kan nu gewijzigd of verbeterd worden. Hiervoor wordt de cursor (het liggende streepje onder de regel) gebruikt. De cursor kan worden bestuurd met de volgende toetsen:

Het liggende dubbele pijltje stuurt de cursor naar rechts, met SHIFT tegelijk ingedrukt naar links.
Het staande dubbele pijltje stuurt de cursor omhoog, samen met SHIFT omlaag.
De cursor blijft automatisch doorlopen als de toets ingedrukt blijft. Dit maakt een snelle positionering mogelijk.
Met (verspringt de cursor naar het begin en met) naar het einde van de regel waarin hij zich bevindt.
Met H verspringt de cursor weer naar zijn beginpositie, linksboven op het scherm.
De plaats waar de cursor zich bevindt kan "gemarkeerd" worden door op de toets @ ("apestaart") te drukken.
Dit wordt aangegeven met een @ op de onderste regel. Het gemarkeerde punt kan worden gewist met de toets ESC.
Als daarna de cursor elders in de tekst geplaatst wordt kan ALLE tekst tussen @ en de cursor met bepaalde commando's gewijzigd worden. Als er geen positie gemarkeerd is met @ wordt alleen de plaats van de cursor beinvloed.

WIJZIGEN OF CORRIGEREN:

Voor het wijzigen of corrigeren van de tekst zijn de volgende commando's beschikbaar::

A - Met A kan tekst tussengevoegd worden achter de huidige plaats van de cursor. Na het intikken van A verschijnt op de onderste regel A: waarachter de tekst getikt kan worden. Met COPY wordt de tekst daarna tussengevoegd.

B - Met B wordt op dezelfde manier tekst tussengevoegd VOOR de cursor.

E - Met E kan tekst worden toegevoegd achter het einde van de tekst, op de plaats van het witte blokje.

DELETE - Als op de knop DELETE gedrukt wordt, dan wordt alle tekst tussen @ en de cursor gewist. Als er geen positie gemarkeerd is met @ wordt alleen de plaats van de cursor gewist.

C - Als op C gedrukt wordt, dan wordt de tekst tussen @ en de cursor gekopieerd in de correctie buffer. Deze tekst kan dan later ingevoegd worden.

T - Met T wordt de tekst tussen @ en de cursor van het scherm verwijderd en in de correctiebuffer opgeslagen voor later gebruik.

R - Als op R gedrukt wordt verschijnt op de onderste regel R: Hierachter ingetikte tekst vervangt, na het drukken op COPY. de tekst tussen @ en de cursor en wordt bovendien in de correctie buffer opgeslagen.

X - Met X wordt het letterteken dat door de cursor aangegeven wordt vervangen door het letterteken dan na X getikt wordt.
Als na X op COPY gedrukt wordt, dan wordt ALLES na de cursor gewist.

I - Met I wordt het letterteken dat na I getikt wordt ingevoegd VOOR de cursor.

ZOEKEN EN VERVANGEN:

1. Zoeken van tekst:

F - Met het commando F kan de tekst op het scherm afgezocht worden naar het voorkomen van een opgegeven stuk tekst (of een woord of letterteken). Vanaf het begin van de huidige schermpagina worden alle plaatsen opgezocht waar de opgegeven tekst voorkomt. Deze kan dan selectief of op alle voorkomende plaatsen weggelaten worden of vervangen door een andere tekst.
Te te vinden tekst moet omsloten worden door een teken dat niet in de te doorzoeken tekst voorkomtn (bijv. / of ! of &). Het zoekkommando gaat er dan bijvoorbeeld als volgt uitzien:

F/Huis/

Als hierna op COPY gedrukt wordt verschijnt het tekstgedeelte dat begint met het gezochte stuk tekst (waar dat het EERST voorkomt) linksboven op het scherm.

2. Zoeken en vervangen:

Als na het te vinden stuk tekst een tweede stuk tekst opgegeven wordt, dat afgesloten wordt met hetzelfde teken, dan kan dit het eerste stuk tekst vervangen. Het F-commando gaat er dan bijv. als volgt uitzien:

/huis/woning/

Als er nu op COPY gedrukt wordt, dan verschijn de tekst die begint met het eerste voorkomen van "huis" linksboven op het scherm.
Beneden op het scherm verschijnt:

F:

Dit gebeurt iedere keer dat het woord "huis" gevonden wordt.
Als hierna Y wordt getikt, wordt op die plaats "huis" vervangen door "woning". Als N wordt getikt, wordt er niet vervangen. Dit gaat dooor tot alle plaatsen waar "huis" stond geweest zijn.
Met ESC kan het F-commando tussentijds uitgeschakeld worden.
Als bet F-commando afgesloten wordt met A of a dan wordt AUTOMATISCH het eerste stuk tekst op ALLE plaatsen vervangen door het tweede. Bijv: F/huis/woning/a vervangt "huis" overal door "woning".
Als het tweede tekstgedeelte "leeg" is, dan kan de te vinden

tekst worden gewist.
Met bijvoorbeeld: F/huis// kan "huis" overal gewist worden.
Meet F/huis//A overal in 1 keer.

WEGSCHRIJVEN OF LADEN VAN "FILES".

Voor het wegschrijven van teksten (files) naar disk of tape wordt het commando > gebruikt.
Na > kan een naam opgegeven worden, bijv:

> BRIEF 8.8.82

Als nu op RETURN gedrukt wordt, dan wordt de gehele tekst naar disk of tape weggeschreven.
Voor het laden van een tekst van disk of tape wordt het commando < gebruikt. bijv:

<NOTULEN

Als hierna op RETURN gedrukt wordt dan wordt het gehele file van disk of tape geladen. Als er met cassette gewerkt wordt dan geeft het Cassette Operating System (COS), nadat op RETURN gedrukt is, het bericht PLAY TAPE of RECORD TAPE. Hierna moet de cassetterecorder aangezet worden, waarna op de spatiebalk gedrukt moet worden. Nu wordt de tekst geladen of weggeschreven.

Tekst die men in de correctie buffer heeft zitten kan in het tekstgeheugen gezet worden door achter de commando's A, B, en E direct de COPY toets indrukken. Geen tekst achter de commando's intypen omdat anders de correctiebuffer veranderd wordt.

DIVERSE COMMANDO'S:

W - Met W kan worden nagegaan hoever het tekstgeheugen (het lage geheugen van de ATOM) gevuld is. Na het intikken van W verschijnt het hexadecimale adres van het einde van de tekst op het scherm. Het tekstgeheugen begint op #2800 en eindigt in een volledig intern uitgebreide ATOM op #4000. Met externe geheugenkaarten kan het tekstgeheugen uitgebreid worden tot #8000.

* - Achter * kan een COS of DOS commando gebruikt worden vanuit de editor. Bijvoorbeeld met :

*CAT

Hiermee kan een catalogus oppgevraagd worden van alle files op een tape of disk. Na het gebruiken van een * commando moet op de spatiebalk gedrukt worden om naar het tekstverwerkings programma terug te keren.
Het toevoegen van tekst van tape of disk aan een al geladen tekst gaat als volgt:
Vraag met W het geheugenadres op van de al geladen tekst en laad de toe te voegen tekst op dat adres met *LOAD. Als bijvoorbeeld na het intikken van W het adres 29AE verschijnt kan het file "deel twee" met

*LOAD "deel twee" 29AE

toegevoegd worden.

Q - Met Q wordt de editor verlaten, De tekstspace-pointer wordt op #82 gezet en de inhoud van het tekstgeheugen wordt in het hoge geheugen (vanaf #8200) van de ATOM geladen, alsof deze vanaf het toetsenbord was ingetikt.
Zo wordt de tekst van een BASIC programma dat met de tekstverwerker is ingetikt automatisch omgezet in een werkend programma in het hoge geheugen. De inhoud van het tekstgeheugen van de editor blijft ongewijzigd, zodat door EDIT in te tikken de tekstverwerker opnieuw aangezet kan worden en het programma eventueel nog verder gewijzigd kan worden.

O - Met O wordt de tekst bewerkt en naar de printer gestuurd en afgedrukt. Daarvoor moeten eerst bewerkings-commando's in de tekst ingevoegd zijn. Deze commando's worden verderop behandeld.

ESC - Met ESC wordt ten alle tijde de uitvoer naar scherm en printer stopgezet.

AANVULLING ED64:

O - Met O krijgt men toegang tot de mogelijkheid om tekst af te drukken op scherm of printer.
Na nogmaals op O gedrukt te hebben wordt de tekst alleen op het scherm gezet, dit ter controle van de tekst voor het naar de printer gestuurd wordt.

P - Met P wordt de tekst (nadat eerst op O gedrukt is) naar de printer gestuurd

SHIFT - Met SHIFT kan de tekst op het scherm stilgezet worden, is niet van toepassing als de printer aanstaat.

CTRL - Met CTRL wordt weer verdergegaan met het afdrukken van de tekst op het scherm.

FOUTMELDINGEN:

Pieptoon - Geeft aan dat de cursor op een ongeldige plaats staat, bijvoorbeeld voorbij het einde van de tekst. Een poging om dan de commando's A, B, of R te gebruiken geeft een pieptoon.
Een pieptoon kan ook betekenen dat @ twee keer tegelijk gebruikt wordt om een plaats in de tekst te markeren. Als dit gebeurt moet op ESC gedrukt worden om de fout op te heffen.

**Press COPY - Als deze melding verschijnt terwijl een van de commando's A, B, R, of T gebruikt is, dan betekent dit dat de correctiebuffer vol is. Door op COPY te drukken wordt teruggekeerd naar de editor en de inhoud van de correctiebuffer bewaard. Deze moet nu eerst in de tekst ingevoegd worden voor doorgegaan kan worden met intikken van tekst.
Als de melding verschijnt terwijl het commando E gebruikt wordt om tekst in te tikken, dan is het tekstgeheugen van de editor vol. De enige oplossing is het wegschrijven van de tekst naar tape of disk en met een nieuwe file beginnen.

**Full - Deze melding geeft aan dat tekst die ingetikt is achter A of B te lang is om nog in het tekstgeheugen erbij te passen. De

inhoud van de correctie buffer blijft bewaard maar wordt niet in de tekst ingevoegd. De inhoud van het tekstgeheugen dient op tape of disk weggeschreven te worden en de inhoud van de correctie buffer overgebracht naar het tekstgeheugen.

**Press SPACE - Iedere poging om tekst weg te halen voorbij het begin van de tekst die achter A, B, E of R ingetikt wordt, laat dit commando vervallen en geeft deze melding. Met een spatie kan weer naar de editor worden teruggekeerd.

Het intikken van niet bestaande commando's en het aanslaan van verkeerde toetsen wordt door het tekstverwerkings programma genegeerd.

BEWERKINGSCOMMANDO'S:

De bewerkingscommando's (processing commands) moeten voor het afdrukken in de tekst worden ingevoegd om de vorm waarin de tekst wordt afgedrukt te bepalen. Ze moeten aan het begin van een zin staan, dus na de RETURN van de vorige zin, of aan het begin van het document.
De commando's bestaan uit een controleteken, dit is in normale gevallen een punt, gevolgd door een enkele kleine letter, en soms gevolgd door een getal. De volgende commando's kunnen gebruikt worden:

.l - Eerste zin.
Dit commando MOET gebruikt worden om het begin van een document aan te geven. Het initialiseert alle bewerkings commando's op hun standaardwaarde (default setting), d.w.z. de waarde die alle commando's automatisch aannemen tenzij een andere waarde opgegeven wordt.

.l en .r - Paginagrootte.
Het commando .l dient ook om het aantal regels tekst per pagina te bepalen. Dit kan door achter .l het gewenste aantal te tikken, bijv: 35
De standaardwaarde is 58 regels per pagina.
Het commando .r dient om de regellengte en dus de plaats van de rechtermarge te bepalen. De standaardwaarde is 70 tekens.

.p .e en .w - Paginanummering.
Als de bladzijde van een document genummerd moeten worden dan kan met .p het gewenste nummer van de beginpagina van het document opgegeven worden. Met .p0 kan de bladzijdenummering weer afgezet worden. De standaard is geen bladzijdenummering.
De nummering staat standaard gelijk met de rechtermarge op oneven en met de linkermarge op even pagina's.
Met .w wordt de breedte, waarover de paginanummering wordt geprint, bepaald . Met de standaardwaarde 70 komt de nummering gelijk met de linker en rechtermarge. Als .w een kleinere waarde heeft dan komt de nummering dichter naar het midden te staan.
Met .e wordt de nummering midden onder de pagina's gezet.
* Pagina nummering werkt enkel als met .l een waarde opgegeven is en .k aangegeven is (.k - per document printen) met als afsluiting na de laatste regel .a.

.j en .n - Uitvullen tekst.

De tekst wordt standaard bij het afdrukken uitgevuld met extra spaties, waardoor de rechtermarge een rechte lijn wordt. Dit uitvullen kan worden afgezet met .n en weer aangezet met .j.

.s en .d -Regelafstand.
De tekst wordt standaard afgedrukt met enkele regelafstand. Met de .d kan op dubbele regelafstand overgegaan worden en met .s weer op enkele.

.i .t en .m -Inspringen.
Een stuk tekst kan men laten inspringen met het commando .i gevolgd door het aantal posities dat de linker marge moet opschuiven. bijvoorbeeld:
.i7 Deze tekst wordt 7 spaties naar rechts afgedrukt. Dit blijft ge-
handhaafd tot het inspringen is afgezet met het commando genaamd .i zonder getal.
Met .t plus een getal wordt een enkele regel ingesprongen, onafhankelijk van een eventueel .i commando. Het .t commando kan dus gebruikt worden om in te springen aan het begin van een nieuwe paragraaf.
Het .m commando staat gelijk aan .t0 ,maar het negeert een getal er achter. Het kan dus gebruikt worden om voor een met .i ingestelde marge nummers af te drukken.

.c - Centreren tekst.
Met .c wordt de tekst die smaller is dan de regellengte gecentreerd. Als er marges verschoven zijn, dan wordt de tekst tussen de marges gecentreerd.

.a - Ruimte maken.
Het .aN commando geeft aan dat de volgende N regels op dezelfde bladzijde moeten komen. Als er niet genoeg ruimte is tot onderaan de bladzijde dat wordt op een nieuwe bladzijde begonnen vanaf het punt waar .aN in de tekst voorkomt. Bijvoorbeeld:

.a4 Lijst van commando's:

zal tot gevolg hebben dat tenminste 3 regels van de lijst onder de kop op dezelfde pagina afgedrukt worden.
Een lege ruimte van 20 regels voor afbeeldingen in de tekst kan gereserveerd worden door het intikken van .a20, met als gevolg 20 lege regels. Een nieuwe bladzijde kan worden begonnen met .a zonder getal.

.k en .o - Besturing printer
De printer drukt standaard de gehele tekst achter elkaar door af, voor gebruik van kettingpapier. Als met losse vellen gewerkt wordt moet .k gebruikt worden. Daardoor wordt per pagina afgedrukt, waarna het bericht:

Paper

op het scherm verschijnt. Na het inzetten van een nieuw vel kan op een willekeurige toets gedrukt worden, waarna een volgende pagina afgedrukt wordt, enz.
Met het .oN commando kunnen speciale "control characters" naar de printer gestuurd worden. Hierbij moet voor N de decimale code van het character ingevuld worden. Deze control characters kunnen per

printer verschillen.

./ - Commentaar.
Een zin die begint met ./ wordt genegeerd bij het afdrukken en kan gebruikt worden om commentaar aan een tekst toe te voegen. Een andere mogelijkheid is het niet laten afdrukken van een stuk tekst zonder dit te wissen. (het schuine streepje ./ moet de andere kant op staan).

.x - Veranderen controleteken.
Met .x kan het controleteken voor bewerkingscommando's, dat normaal een punt is, veranderd worden in een willekeurig ander teken. Bijvoorbeeld: .x% verandert het controleteken van . in % en %x. verandert het dan weer terug in een punt.
Aan het begin van een nieuwe "file" is het controleteken steeds een punt.

BEWERKEN VAN LANGE DOCUMENTEN

Documenten die te lang zijn om in het tekstgeheugen van de Atom te passen dienen in "files" van een geschikte lengte onderverdeeld te worden, die apart op tape of disk bewaard worden. De correctiebuffer wordt niet beinvloed door het laden of wegschrijven van tekst en kan dus gebruikt worden om tekst uit het ene "file" in het andere over te brengen.
Om het hele document af te drukken kunnen de "files" na elkaar geladen, bewerkt en afgedrukt worden. Hierbij blijven de bewerkingscommando's die aan het begin van het EERSTE file gebruikt zijn geldig, tenzij ze veranderd worden.

BEWERKEN VAN BASIC PROGRAMMA'S

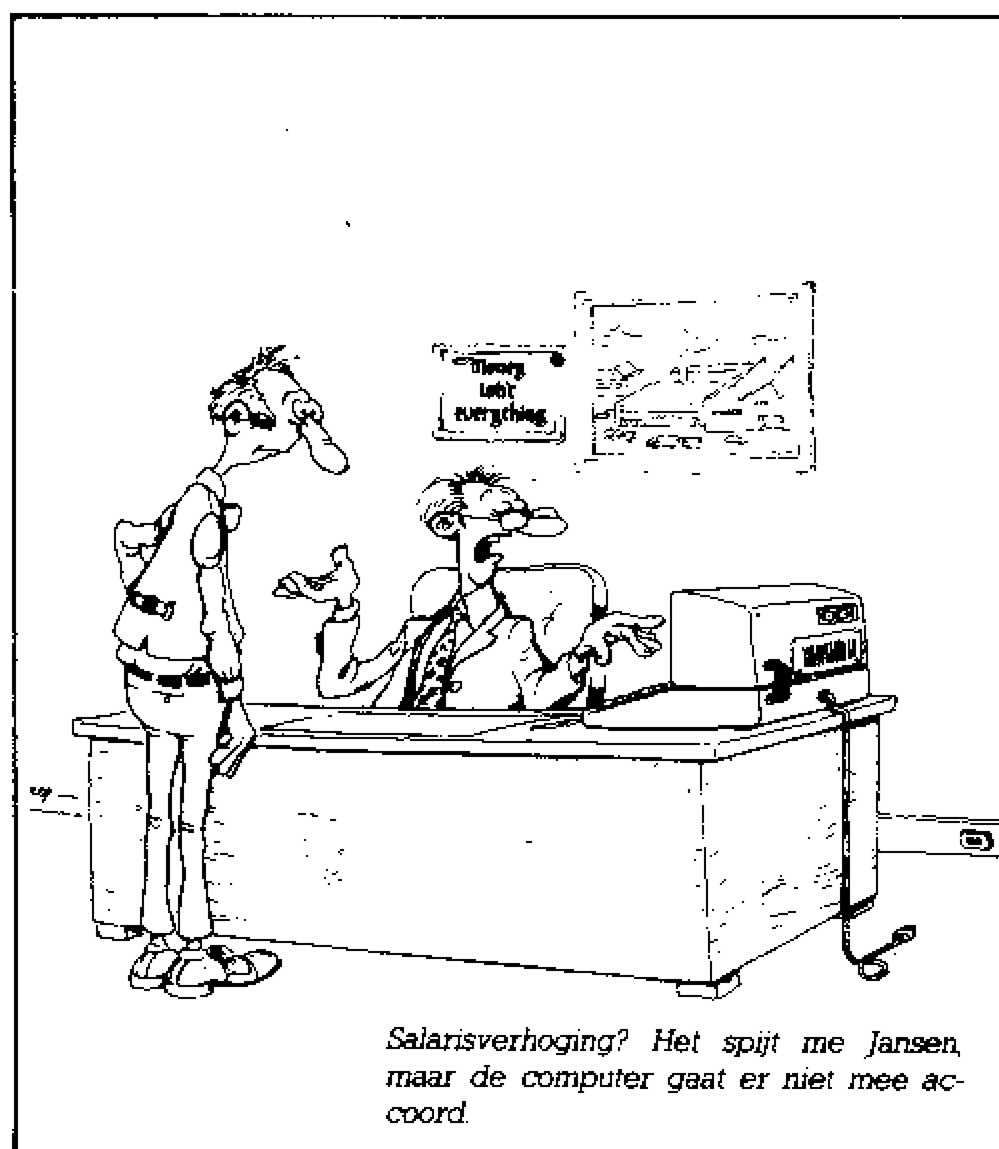
Het TEXT commando.
Om BASIC programma's te kunnen bewerken dienen deze eerst in de goede vorm omgezet te worden. Dit gaat als volgt in zijn werk:
Laad het programma, in dit voorbeeld PROG genaamd, in het hoge tekstgeheugen door de volgende commando's in te tikken:

```
?18=#82
NEW
LOAD "PROG"
```

Hierna dient het TEXT commando aangeroepen te worden door TEXT te tikken. Als er geen Floating point ROM in de Atom zit dient het aangeroepen te worden met LINK #ACFA.
Het commando TEXT zorgt ervoor dat alles wat op het scherm afgedrukt wordt in het tekstgeheugen terechtkomt, alsof het was ingetikt.
Als dus na het aanroepen van TEXT het programma "gelist" wordt met het commando LIST, dan komt het programma in de juiste vorm in het tekstgeheugen terecht. Nu kan met EDIT de tekstverwerker aangezet worden. Op de vraag Old Text? dient nu met Y (Yes) geantwoord te worden. Hierna kan het programma bewerkt worden.
Tenslotte kan het gewijzigde BASIC programma met het commando Q weer in het hoge geheugen van de Atom geladen en geprobeerd worden. BELANGRIJK: Hierbij worden geen programmaregels van meer dan 64 characters lengte geacccepteerd.

Het aanroepen van de tekstverwerker vanuit BASIC.

De tekstverwerker kan worden aangeroepen vanuit een BASIC programma in het hoge geheugen van de Atom door in het programma LINK #ACF7 uit te voeren. Op deze manier kunnen "text files" gemaakt worden of gewijzigd door een BASIC programma met gebruikmaking van de faciliteiten van de tekst verwerker.



Salarisverhoging? Het spijt me Jansen, maar de computer gaat er niet mee accoord.

OVERZICHT COMMANDO'S

Invoer commando's:

Toets:	wijzigt: tekst	buff.	beschrijving:
A	x	x	Invoegen na
B	x	x	Invoegen voor
C		x	KOPIE tekst naar buffer
COPY			Einde tekst
CTRL		ED64	Start uitvoer scherm
DELETE	x		Wissen
E	x		Intikken tekst
ESC			Wissen markering
F			Zoeken (en vervangen)
H			Cursor linksboven
I			Invoegen tekst
N			Volgende schermpagina
O			Afdrukken op printer
O		ED64	Afdrukken op scherm
P		ED64	Na O afdrukken op printer
P			Vorige schermpagina
Q			Naar BASIC programma
R	x	x	Vervangen tekst
S			Naar begin tekst
SHIFT		ED64	Stopt uitvoer scherm
T		x	Tekst naar buffer
W			Adres einde tekst
X	x		Wijzigen teken
Z			Naar einde tekst
^			Tekst een regel omhoog
[			Cursor naar begin regel
]			Cursor naar einde regel
@			Markeren cursorpositie
<			Laden "tekstfile"
>			Wegschrijven "tekstfile"
*			Uitvoeren COS/DOS commando

Bewerkings commando's

.a	nieuwe bladzijde
.c	centreren regel
.d	dubbele regelafstand
.e	paginanummers in het midden
.iN	inspringen met N spaties
.j	uitvullen volgende regels
.k	per pagina afdrukken
.lN	N regels per pagina
.l	eerste regeldocument
.m	marge gelijk aan .tØ
.n	niet uitvullen volgende regels
.oN	control character N naar printer
.pN	paginering begint met nummr N
.pØ	geen paginering
.rN	regellengte naar N tekens
.s	enkele regelafstand
.tN	inspringen 1 regel met N spaties
.w	breedte paginanummers
.x?	veranderd controleteken in teken ?
./	commentaar - wordt niet afgedrukt

Wordpackverandering 3

In Acorn-nieuws jaargang 3, nummer 6 (die met vier eikels voor het jaargangnummer) staat op bladzijde 80 en 81 een aanpassing voor de wordpack beschreven. Daarmee wordt het mogelijk om in een regel control-codes naar de printer te sturen. Dat geldt echter voor de standaard versie en niet voor ED64.
Met de hier beschreven programma kan ED64 worden aangepast, zodanig dat control-codes binnen een regel naar de printer kunnen worden gestuurd. Door een routine te vervangen door een kortere en de aanmeldingstekst te verkorten is er plek gevonden voor wat nieuwe mogelijkheden.
Die mogelijkheden zijn deze:

- .h - Hiermee geef je op welk karakter wordt geprint als spatie. Bv.".h$" aan het begin van een regel doet de editer elke $ vervangen door een spatie. Dit is voor gebruik bij het uitvullen, om te voorkomen dat de edditer op die plaats ekstra spaties plaatst, of er de regel afbreekt.
  (.h van hulpspatie).
- .{, .¦, .}, .~ - Hiermee wordt opgegeven welke escape-sequence gekozen wordt. Als dan in de tekst een shift[(dus {) wordt gevonden, dan wordt de escapecode die daarmee in een tabel wordt aangewezen naar printer en scherm gestuurd. Deze vier karakters kunnen zo ieder een betekenis krijgen. De ruimte die zij in de tekst innemen (1byte) wordt vervangen door een spatie, zodat het uitvullen op de printer goed blijft gaan.
  Het nummer dat achter .{, .¦, .} of .~ wordt opgegeven is het toegangsnummer in de tabel. Alles vanaf die plaats, tot en met het eerste nummer groter dan #7F wordt dan naar de printer gestuurd. Wordt echter 0 opgegeven, dan wordt het oorspronkelijke karakter geprint (dus {¦}~). Met het kommando .l wordt 0 geinitieerd, zodat de default is als de standaard ED64.
  De tabel staat in de listing ingevuld, (regel 1000 t/m 1250), voor een Epson MX80 III, maar kan eenvoudig worden aangepast, en er is nog ruimte voor uitbreiding.
  Om de tabel kort te houden, wordt door het programma de eerste escape al direkt geprint, nodig of niet, en het laatste karakter is (ascii + #80). (Die #80 wordt door de printer toch niet gezien).
  De mogenlijkheden met deze tabel zijn:

funktie	on	off
standaard ascii	0	-
shift out	1	2
shift in	3	4
emphasized	5	6
doubled	7	14
superscript	8	12
subscript	10	12
underline	15	17
emph.+double	19	22
alles uit		22

  Bij gebruik van een andere printer moet misschien de tabel worden aangepast.

En zo is dit verwezenlijkt.
De oorspronkelijke routine die voor de punt-kommando's zorgt werkte met een lijst met compares. (Zie #A8E6 en verder.) Door hiervoor in de plaats een routine met een tabel op te nemen kon ruimte worden gemaakt voor de uitbreiding van dat aantal punt-kommando's. (Basic-regel 130 t/m 730). De tabel bevat het lage byte van het startpunt van de routine die door het puntkommando wordt bedoeld. Het hoge byte is telkens #A9. Door PHA in regel 160 wordt het adres op de stack gezet en door RTS wordt er dan naar toe gesprongen.
De tabel wordt ingevuld in regel 750 t/m 770. Zorg er voor dat er geen label (LLx) ontbreekt, want dan komt er een onbedoeld adres in de tabel. Deze routine met de tabel neemt evenveel ruimte in als de oorspronkelijke routine, na toevoeging van de nieuwe commando's.
Het tweede dat is gebeurd is het opvangen van de output-routine (net als in AN 3-6). Daarvoor is de aanmeldings tekst sterk verkort (zie #AF89 ev.) en op de vrijgekomen plaats een routine gezet. Daarin wordt eerst gekeken of het te printen karakter de spatie-vervanger is, zoja P." ";klaar. Dan wordt gekeken of het karakter {,¦,} of ~ is, zonee print het karakter.
Is het wel een van die karakters, haal dan uit adres resp. #7B,7C,7D,7E vanaf welk karakter uit de escape tabel moet worden geprint, zoals eerder met .{, .¦, .} en/of .~ is opgegeven. Is het een nul; print dan {,¦,} of ~. Is het geen nul; P." ",$27 en print verder uit die tabel tot het laatste karakter is geprint (karakter groter dan #7F).

Ik ben er van uitgegaan dat niemand de listing intypt en heb dus met ED64 wat extra opmerkingen bijgevoegd. Die staan telkens aan het einde van de regel na een \.

```
   10     \ aanpassing van ed64
   20 A=#A000;B=#4000;EPROM 4    \ of waar ED64 nu staat
    30 FOR N=0TO#FFF S.4;B!N=A!N;N.    \ copieer  ED64 naar #4000
ev.
   40 EPROM 0;     \ ram
    50 FOR N=0TO#FFF S.4;A!N=B!N;N.    \ copieer  ED64 naar #A000
ev.
   60 DIM LL255   \ dat zal wel genoeg zijn
   70 FOR N=0TO 255;LLN=-1;N.
    80 T=#AFFF;G=#A9DF     \ T=tabelwijzer, G=de  ED64-routine die
het getal achter het puntkommando leest
   90 P.$21
  100 FOR N=0TO1
  110 P=#A8E6
  120[
  130:LL0 \vervang #A8E6 ev.
   140 INY;LDA (#52),Y;BMI LL50;CMP  @#5C;BCC LL50     \(#52),Y is
het karakter dat  onderhande is. Hier wordt  gekontroleerd of het
niet buiten de tabel valt
  150 STA #21;TAX    \ sla  op in #21 voor later gebruik en zet in
X als tabel-offset
  160:LL51 LDA  @#A9;PHA;LDA T,X;PHA;SEC;RTS    \ zet  #A9xx op de
stack (xx komt uit de tabel) en RTS er naartoe
  170:LL59    \tabelwijzer, de tabel wordt ingevuld  in regel 740
t/m 760
  180]
  190 T=LL59-#5C
  200 P=LL59+35    \ eerste plaats na de tabel
  210[
Nu volgen  eerst de  labels voor  de ascii-waardes  die niet  als
puntkommando voorkomen, maar wel binnen de tabel vallen
  220:LL3     \.]
  230:LL4     \.^
  240:LL5     \.-
  250:LL6     \.'
  260:LL8     \.b
  270:LL12    \.f
  280:LL13    \.g
  290:LL23    \.q
  300:LL27    \.u
  310:LL28    \.v
  320:LL31    \.y
  330:LL32    \.z
  340:LL50    \geen puntkom.
  350 PLA;PLA;LDY @0;JMP #A831
  360:LL 9 ROR #57;RTS; \.c centreer de regel
  370:LL16 STX #8F;RTS; \.j uitvullen
  380:LL20 ROR #8F;RTS; \.n niet uitvullen
  390:LL10 ROR #8E;RTS; \.d dubbel spatie
  400:LL25 STX #8E;RTS; \.s single spatie
   410:LL15  JSR G;LDA  #8B;ADC  #8D;SEC; STX  #8D;BCS LL40;  \.i
inspringen
  420:LL24 JSR G;SEC;    \.r rechter kantlijn
  430:LL40 SBC #8D;STA #8B;RTS
  440:LL21 JSR G;JMP #FFF4; \.o output karakter
  450:LL26 JSR G;          \.t tijdelijk inspringen
  460:LL45 LDA #8B;STA #5B;ADC #8D;STX #5A
  470 SEC;SBC #5A;STA #8B;RTS
```

```
480:LL17 ROR #91;RTS; \.k keypress voor volgende pagina
490:LL41 LDX #88;BEQ LL42
500:LL43 JSR #FFED;DEX;BNE LL43
510:LL42 JMP #AA25
520:LL29 JSR G; \.w wijdte voor pagina nummering
530:LL48 STA #8C;RTS
540:LL22 JSR G;STX #89; \.p pagina nummer
550:LL47 LDA @0;STA #8A;RTS
560:LL7 JSR G;BEQ LL41;CMP #88;BEQ LL44;BCS LL41; \.a allow new
page of zoveel regels op dezelfde pagina's
570:LL44 RTS
580:LL19 LDX @0;CLC;BCC LL45; \.m marge 0 (=.t0)
590:LL18 JSR G;BNE LL46;LDA @#3A; \.l line 1 en hoeveel volgen,
initieer
600:LL46 STA #90;STA #88;JSR LL47;LDX @#4
610:LL33 STA #7A,X;STA #8B,X;DEX;BNE LL33
620 STA #91;STA #92;STA #89;STA #7F
630 LDA @#46;STA #8B;BNE LL48
640:LL30 INY;LDA (#52),Y;STA #5F;RTS ; \.x vervang . door ander
650:LL 2 LDA @#0D; \.\ wat volgt niet printen (kommentaar)
660:LL49 INY;CMP (#52),Y;BNE LL49;RTS
670:LL11 ROR #92;RTS; \.e paginanummers in het midden van (.w)
680:LL14 INY;LDA (#52),Y;STA #7F;RTS \.h hulpspatie : nieuw
690:LL33   \.{ : nieuw
700:LL34   \.¦ : nieuw
710:LL35   \.} : nieuw
720:LL36 JSR G;LDX #21;STA #00,X;RTS \.~ : nieuw
730:LL90;];      \ maximaal #A9D5.  Door PRINT LL90  kan worden
gekontroleerd of de routine (met tabel ) niet te lang is geworden
740 N.;P=LL59;FOR N=0 TO 34
750 P?N=(LL(N+2)%256)-1     \ hier wordt de  tabel ingevuld. Het
low-byte wordt berekend -1, omdat bij  een RTS de stack-waarde +1
in de programcounter wordt gezet
760 N.
770 GOS.o    \ daar wordt de aanmeldings-tekst verkort neergezet
en LL72 komt terug met de eerste vrije plaats na die tekst
780 FOR N=0TO1
790 P=LL72
800[
810:LL52 CMP #7F;BEQ  LL60;  \.hCHAR.  er moet dus  een spatie
geprint worden
820 CMP @#7B;BCC LL61 (print). Het is niet {,¦,} of ~ dus print
het karakter
830 TAX;LDA #0,X;BEQ LL62;TAX      \  haal het getal op  dat met
.{,.¦,.},.~ of .l is opgegeven en zet dat in X als tabeloffset
840 JSR LL60 \vul op met spatie
850 LDA @#1B \escape
860:LL63 JSR #FFF4   \ outputroutine
870 LDA LL65,X;BMI LL61;INX;BNE LL63 \loop tot char.>#7F
880:LL60 LDA @#20  \ space
890:LL61 JSR #FFF4 \ print
900 RTS
910:LL62 TXA;BNE LL61
920:LL65
930]
940 N.
950 ?#A8CC=LL52%256    \
960  ?#A8CD=LL52/256   \ verander  in ED64 de JSR  #FFF4 in JSR
LL52. M.a.w. vang de outputroutine op
```

```
  970 P.$6
  980 P=LL65
```

Hierna wordt de printer-afhankelijke tabel ingevuld

```
  990 P?0=0   ;    \ gewone char.
 1000 P?1=#8E ;    \ shift out on
 1010 P?2=#94 ;    \ shift out off
 1020 P?3=#8F ;    \ shift in on
 1030 P?4=#92 ;    \ shift in off
 1040 P?5=#C5 ;    \ emphas.on
 1050 P?6=#C6 ;    \ emphas.off
 1060 P?7=#C7 ;    \ double on
 1070 P?8=#53 ;    \ superscr.on
 1080 P?9=#80
 1090 P?10=#53 ;    \ subscr.on
 1100 P?11=#81
 1110 P?12=#54 ;    \ s/s-scr.off
 1120 P?13=#1B
 1130 P?14=#C8 ;    \ double off
 1140 P?15=#2D ;    \ underl.on
 1150 P?16=#81
 1160 P?17=#2D ;    \ underl.off
 1170 P?18=#80
 1180 P?19=#45 ;    \ emp+doubl.on
 1190 P?20=#1B ;P?21=#C8
 1200 P?22=#46 ;    \ all off
 1210 P?23=#1B ;P?24=#54
 1220 P?25=#1B ;P?26=#48
 1230 P?27=#1B ;P?28=#2D ;P?29=0
 1240 P?30=#14 ;P?31=#92
 1250 E.
 1260o P=#AF98;$P="OLD TEXT Y/N"
 1270 P=P+L.P;?P=#EA;P?1=#60
 1280 LL72=P+2
 1290 R.
```

Na het runnen van dit programma kan het hele A-blok worden geprogrammeerd in een EPROM en verder zijn diensten bewijzen.

PS. Bij zeer intensief gebruik van deze ED64TR (een werkstuk met zo'n 200 pagina's tekst) is gebleken dat na het gebruik van de floppy vanuit de editter, het zero-page adres #20, dat ik eerst gebruikte voor .h, werd overschreven. Daarom ben ik overgegaan naar #7F. Of deze en/of andere toegepaste zero-page adressen problemen geven moet nog gaan blijken. Tot nu toe lijkt het allemaal goed te gaan, maar je bent gewaarschuwd.

Zero-page gebruik:
#21 tijdelijke opslag van de puntkommando letter.
#7B tabel-pointer zoals opgegeven met .{. Na .l staat er 0.
#7C tabel-pointer zoals opgegeven met .|. Na .l staat er 0.
#7D tabel-pointer zoals opgegeven met .}. Na .l staat er 0.
#7E tabel-pointer zoals opgegeven met .~. Na .l staat er 0.
#7F het karakter dat bij uitprinten wordt vervangen door een spatie, zoals opgegeven met .h. Na .l staat er 0.

Het MACINTOSH programma

Dit is een uitgebreid teken programma wat gebaseert is op het gebruik van een joystick in plaats van een muis. Met de joystick wordt getekend en worden de verschillende mogelijkheden gekozen. Dit is ongeveer overeenkomstig met het gebruik van een muis vandaar de naam macintosh. Het enige nadeel is voor de meeste mensen dat het programma vrij groot is en dat verschillende boxen gebruikt worden. Benodigd zijn:

- p-charme
- gagsrom
- geheugen van #2500 tot #6800

Op #2500 staat een flow routine.
Op #2600 staat het shape commando.
Op #2700 zit een shape tabel (kruis).
Op #2710 staat het move commando.
En op #2720 staat een joystick uitlees routine.

Het gebruik:
Laad het programma met *LOAD "MACINT" en run het na een END. Het programma MOET op #2900 gerund worden. Nu ziet u dat er een scherm wordt gemaakt met aan de randen aparte bloken met commmando's en in het midden staat een kruis. Door nu de joystick te bewegen zal het kruis over het beeldscherm verschuiven. Als u nu met het kruis bijvoorbeeld naar CAT gaat en dan binnen dit blok op de vuurknop drukt van de joystick dan wordt dit commando uitgevoerd. (Er komt dan een disk catalogus). De rij commando's die boven aan het scherm staan spreken eigenlijk voor zich. De onderste rij zijn teken commando's en de verticale rijen zijn paint patronen. De mogelijke commando's zijn als volgt:

De bovenste rij:

INV - inverteer het scherm
NEW - wis het scherm en begin opnieuw
CAT - maak een disk catalogus
OFF - zet alle commando's af en ga naar het midden van het scherm
SAVE - save het beeldscherm op de diskette
LOAD - load een eerder gesaved scherm
QUIT - stoppen

de onderste rij:

CIRKEL - teken cirkels
POTLOOD - tekenen als met een potlood
M/D - het move/draw commando
FILL - patroon opvul routine
CL. - uitwissen van fouten
? - compacte instructie uitleg

Het werkt verder als volgt. Als aanwijs punt wordt het meest linkse puntje van het krusje gebruikt. Dit is om programmeer technische en gebruikers redenen gedaan. Ik vond in practijk dat dit makkelijker was als het middelpunt van het kruisje. Voor het tekenen van een cirkel gaat u eerst naar de cirkel op de onderste rij en drukt dan op de knop als het kruis zich helemaal binnen

het vak bevindt. U hoort dan een geluidje. Nu is dus het cirkel teken commando aangeroepen. U gaat nu eerst naar de plaats waar het middelpunt van de cirkel moet komen en laat dan de joystick los en drukt op de knop. U hoort een lage pieptoon en er wordt een puntje geplot om aan te blijven geven waar het middelpunt is gekozen. Nu gaat u de rand van de cirkel aangeven en drukt dan weer op de knop en zie de cirkel staat er. Het MOVE/DRAW commando werkt al haast net zo. U gaat eerst weer het commando kiezen en drukt dan op de knop waar het het begin van de lijn moet komen. Ook nu hoort u weer een lage pieptoon. Vervolgens gaat u naar het eindpunt van de lijn en drukt alweer op de knop. Als het goed is wordt er nu een lijn getrokken. Het POTLOOD commando werkt zo, om te tekenen moet u de knop indrukken en onderwijl de joystick bewegen. Fouten kunt u met de CL. routine herstellen. Kies het commando en ga naar de fout toe. Als u nu op de knop drukt wordt een vierkantje ter grootte van het kruis schoon gemaakt. Kleine foutjes kunt u met de POTLOOD routine uitwissen omdat deze inverterend werkt. Met het FILL commando kunt afgesloten vlakken opvullen met patronen uit de verticale vakken. Eerst het FILL commando aanroepen en dan naar het vlak gaan met het te kiezen patroon, ook in dit vlak weer op de knop drukken. Als u nu klaar bent met tekenen gaat u naar SAVE en er wordt dan gevraagd hoe u de tekening wil saven met of zonder menu rand. Dit is natuurlijk nodig als u uw tekening voor een ander programma wil gebruiken. Als u later de tekening zonder menu rand later weer van schijf haalt wordt er getest of er een menu rand is of niet. Zo niet dan wordt er opnieuw een menu rand gemaakt. Voor een ding moet u wel oppassen, u mag de vuurknop van de stick niet indrukken als u de stick beweegt. Een uitzondering hierop zijn de CL.- en het POTLOOD commando's. Ook moet u ermee rekening houden dat het meest linkse puntje van het kruis gebruikt wordt als aanwijspunt. Ik hoop verder dat u er wat aan heeft en veel plezier ermee.

Tenslotte nog over het programma zelf. Allereerst wordt er een scherm met commando's aangemaakt waarna er een kruisje geplot wordt met het shape commando (komt uit de CX-3.02 box). Vervolgens wordt er steeds naar de joystick gekeken waarna het kruisje verplaatst wordt. Komt het kruisje in de buurt van de commando's dan wordt er gekeken welk commando aangeroepen moet worden. Voor het saven, laden en het maken van een catalog wordt het scherm op
geslagen in het gebied van #5000 tot #6800. Nadat het commando is uitgevoerd wordt het opgeslagen scherm weer terug gekopieerd. Het programma was bij het schrijven zeer overzichtelijk maar ik heb veel procedures af moet korten omdat er toch nog iets te weinig geheugen voorradig was. Hierdoor is het wel veel minder leesbaar geworden maar daar tegenover staat dat het wel weer sneller is geworden. Het programma werkt namelijk ook nog recursief waardoor er nog extra geheugen nodig is. En hierdoor werden de patroon tabellen aangetast. De patronen zijn eventueel zeer snel aan te passen omdat ze in het programma staan en hebben de nummers van 1 tot 12. Het programma is eventueel nog verder uit te breiden. Zoals het nu is worden er 27 procedures gebruikt en blijven er voor uitbreiding nog 6 over. Ik zelf denk nog aan een continue opgave van de coordinaten op interrupt maar daarvoor moet ik nog wat meer in de machinetaal duiken.

Dick van den Ende, Dordrecht.

ANIMATIE

Door Robbie Niessen en Bernard Rutgrink

Animatie is een tekenprogramma om tekenfilms te maken. Men heeft minimaal geheugen nodig van #1000 tot #1300 en van #2900 tot #3C00. Met dit geheugen kan je maximaal 50 tekeningen opslaan. Het opslaan van 1 tekening kost 24 bytes dus als je geheugen van #2900 tot #8000 loopt kan je 730 tekeningen opslaan. Voor dit progamma heb je de AXR 1 en P-CHARME rom nodig. Deze moeten door elkaar gebruikt kunnen worden.
Na het starten van het programma verschijnt het menu. Vanuit hier kunt u kiezen uit de volgende dingen.

1) start film
2) edit film
3) load film
4) save film
5) stop

START FILM.
De film die in het geheugen zit wordt dan vertoont. Eerst moet u een paar vragen beantwoorden. Bij de eerste vraag moet u de vertraging invoeren bv.50. Daarna wordt er gevraagt of u de tekeningen aan een stuk door wilt zien. Als u antwoord met nee dan komt de volgende tekening pas als u een toets heeft ingedrukt. Hierna moet u invoeren of u de hele film wilt zien of maar een deel van de film. Wilt u maar een klein deel zien dan vraagt hij van waar tot waar u de film wilt zien.

EDIT FILM.
Dit is het belangrijkste deel van het programma. Vanuit hier kunt u een nieuwe film maken of een bestaande wijzigen. Om te beginnen vraagt de computer welk tekeningnummer u wilt zien. Als er nog geen film is dan begint u bij nummer 0. De tekening van het mannetje bestaat uit 12 co-ordinaten. Elke co-ordinaat komt over een met een lichaamsdeel (zie tek.). U kunt deze veranderen door op de 'L' te drukken. De computer vraagt dan een lichaamsdeel nummer (0-11). U voert dit in en u kunt de nieuw gekozen co-ordinaat verschuiven. Het verschuiven doet u met de volgende toetsen.

Z : links
X : rechts
[: boven
/ : beneden

Met de 'rept' toets ingedrukt kunt de beweging versnellen. Met de 'T' kunt u naar een ander tekening nummer. Met de 'B' kunt u de bewegingen tussen een aantal tekeningen laten berekenen. Als u bv. tekening nummer 0 en 5 heeft getekent dan kunt u hiermee de tussenliggende tekeningen berekenen. De computer tekent nu zelf de tekening nummers 1 - 4. Als je dit later ziet in een film dan zie je dat hij van tekening nummer 0 geleidelijk overgaat naar nummer 5. Dit is dus een makkelijk hulpmiddel voor als u een lange film wilt maken. Met de 'S' gaat u weer terug naar het menu.

LOAD FILM.
Hiermee kan je een film laden.

SAVE FILM
Hiermee save je de film die in het geheugen staat

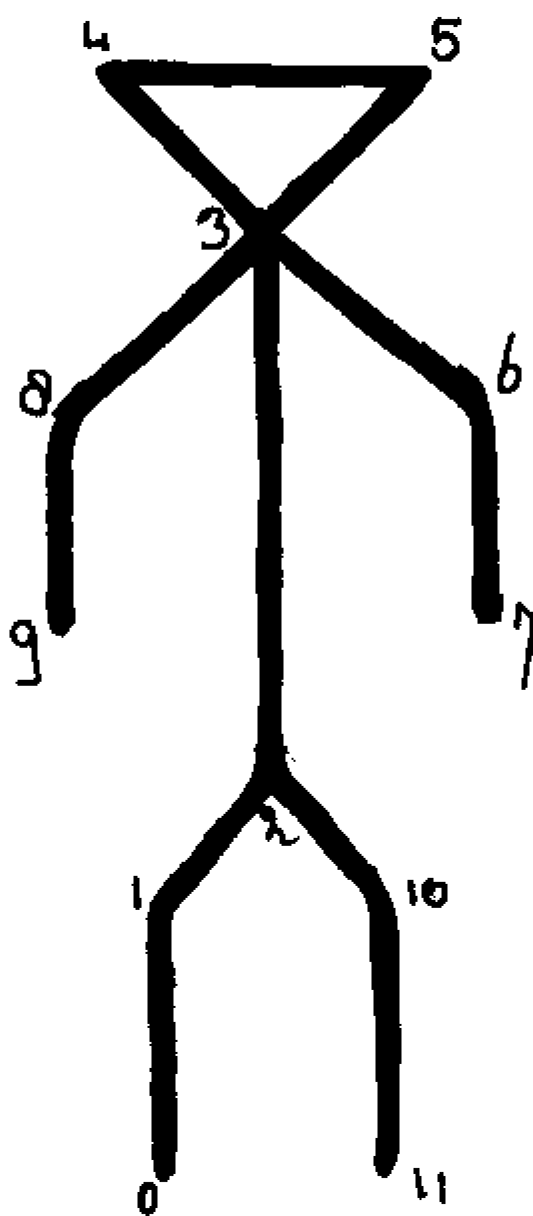
STOP
Beeindigt programma.

```
   10 PROGRAM ANIMATIE
   20 REMMADE BY HAMBURGERSOFT
   30 RELOC #F546,#F73B,#1000
   40 BEEP 1,1;@=0
   50 DIM MM4
   60 F.Z=1 TO 4;MMZ=-1;N.
   70 ON ERROR P.$7'"FOUTE INVOER";LI.#FB7A;G.1970
   80
   90 PROC MENU
  100 GR.;?#E1=0;P." ";HTAB10
  110 P."* ANIMATIE *"';HTAB8
  111 P."BY HAMBURGERSOFT"
  120 P.''''"1:START FILM"'
  130 P."2:EDIT FILM"'
  140 P."3:LOAD FILM"'
  150 P."4:SAVE FILM"'
  160 P."5:STOP"''"UW KEUS ?"
  170 INKEY G;G=G-48
  180 IF G=1;FILM
  190 IF G=2;EDIT
  200 IF G=3;LEES
  210 IF G=4;SCHRIJF
  220 IF G=5;P.$12"OK"'';E.
  230 IF G<1 OR G>5;G.170
  240 G.100
  250 PEND
  260
  280 PROC ZETMACH
  290 FOR Q=1 TO 2;P=#1069
  300 P.$21;[;JSR #F7AA;RTS
  310];P=#10C6;[;RTS;RTS;RTS;]
  320 P=#117A;[;RTS;RTS;RTS;]
  330 P=#11F5;[
  340:MM0
  350 LDX @0
  360:MM1
  370 INX
  380 LDA #700,X
  390 CMP @100
  400 BEQ MM2
  410 LDA @5
  420 STA #16
  430 CPX @100
  440 BNE MM3
```

```
450:MM2
460 LDA @4
470 STA #16
480 INX
490:MM3
500 LDA #700,X
510 TAY
520 LDA(#84),Y
530 STA #5A
540 LDA @0
550 STA #5B
560 STA #5D
570 INY
580 LDA(#84),Y
590 STA #5C
600 STX #9F
610 JSR #102A
620 LDX #9F
630 CPX @18
640 BNE MM1
650 RTS
660]
670 NEXT Q;P.$6
680 PEND
690
710 PROC FILM
720 CLEAR4;P.$30
730 F=0;C=0;L=N;S=2;D=0
740 DO
750 IN.''''"VERTRAGING          "D
760 U.D>0
770 IN.''"AUTO REPEAT (J/N)"$A
780 XIF $A="J";S=0
790 ELSE S=1
800 IN.''"TOTALE FILM (J/N)"$A
810 IF $A="J";G.870
820 DO;CLEAR4
830 P.$30"AANTAL TEKENINGEN "N'
840 IN.'"VAN "F
850 IN.'"TOT "L
860 U.F>0 AND F<L AND L<N
870 C=F;F=B+F*24
880 L=B+L*24
890 J=1;CLEAR 4
900 FOR Q=F TO L STEP 24
910 CLEAR 4
920 !#84=Q;LI.MM0
930 P.$30,C
940 J=3-J;C=C+1
950 IF S=1;INKEY Z;G.970
960 FOR Z=1 TO D;N.
970 NEXT Q
980 LI.#FFE3
990 PEND
1000
1020 PROC EDIT
1030 ?#E1=0;DO;DO
1040 P.''"AANTAL TEKENINGEN-"N'
1050 IN."WELKE TEKENING "M
```

```
1060 U.M>=0
1070 P=B+M*24
1080 IF P>=E;HOOGTE
1090 S=0;Q=P-24
1100 DO;CLEAR 4
1110 X=P+S*2;%X=?X
1120 Y=P+S*2+1;%Y=?Y
1130 !#84=P;LI.MM0
1140 P.$30"TEKENING NUMMER-"M'
1150 INKEY C;Q=2;IF?#B002&#40=0;Q=4
1160 IF C=CH"X";%X=%X+Q
1170 IF C=CH"Z";%X=%X-Q
1180 IF C=CH"[";%Y=%Y+Q
1190 IF C=CH"/";%Y=%Y-Q
1200 IF %X>255;%X=255
1210 IF %Y>190;%Y=190
1220 IF %X<0;%X=0
1230 IF %Y<0;%Y=0
1240 ?X=%X;?Y=%Y
1250 IF C=66;BEREKEN
1260 IF C=76;DEEL
1270 U.C=84 OR C=83
1280 U.C=83;P.$12
1290 PEND
1300
1310 PROC ZETVAR
1320 B=TOP+99;RESTORE 1710
1330 FOR Q=#701 TO #712;READ W;?Q=W;NEXT Q
1340 FOR Q=B TO B+23;READ W;?Q=W/4;NEXT Q
1350
1360 E=B+24;N=0;D=0;!#84=B
1370 A=#2800;$A=""
1380 PEND
1400
1410 PROC BEREKEN
1420 P.$30';DO
1430 IN."BEREKEN"'''"VAN "F
1440 IN."TOT "L
1450 U.F<L AND F>=0 OR L>N
1460 P."EVEN GEDULD"
1470 I=L-F
1480 F=B+F*24;L=B+L*24
1490 FOR Q=0 TO 23
1500 D=(L?Q-F?Q);%D=D/I;C=0
1510 FOR W=(F+24) TO (L-24) STEP 24
1520 C=C+1;Z=F?Q;%Z=Z+(C*%D)
1530 Z=%Z;W?Q=Z
1540 NEXT W
1550 NEXT Q
1560 PEND
1570
1580 PROC HOOGTE
1590 FOR Q=0 TO 23
1600 P?Q=E?(Q-24)
1610 NEXT Q
1620 E=P+24;N=M
1630 PEND
1640
1650 PROC DEEL
```

```
1660 DO;P.$30
1670 IN."NIEUW LICHAAMSDEEL (0-11) "S
1680 U.S>=0 AND S<12
1690 PEND
1700
1710 DATA 100,0,2,4,6,8,10,6,12,14
1720 DATA 100,6,16,18,100,4,20,22
1730 DATA 450,000,450,150,500,200
1740 DATA 500,400,450,500,550,500
1750 DATA 600,350,500,250,400,350
1760 DATA 500,250,550,150,550,000
1770
1780 PROC LEES
1790 P."LADEN"'';X=#2800
1800 IN."WAT IS DE TITEL "$X
1810 F=FIN $X
1820 F.Z=B TO GET F
1830 ?Z=BGET F;N.Z;E=Z-1
1840 N=((E-B)/24)-1
1850 *SHUT F
1860 PEND
1870 PROC SCHRIJF
1880 P."SAVEN"'';X=#2800
1890 IN."WAT IS DE TITEL "$X
1900 F=FOUT $X;PUT F,E
1910 F.Z=B TO E
1920 BPUT F,?Z;N.
1930 *SHUT F
1940 PEND
1950 ZETMACH
1960 ZETVAR
1970 MENU
1980 E.
```



TITRATIE

Na een aantal half afgemaakte probeersels tot het maken van een simulatie-programma, is er eindelijk eentje af.

Voor dit programma is nodig:

* P-Charme ROM
* GAGS ROM
* Geheugen van #2900 tot #6000
* Goede schakelsoftware, zoals BRANQUART, CK, e.a.

Ondanks het feit dat dit een scheikundige simulatie is, hoeft men geen scheikundige kennis te hebben om het te laten werken of om er iets van te begrijpen.
Voor het begrip een korte uitleg, waarbij zo min mogelijk scheikundige begrippen ter sprake komen.
Een titratie dient om een bepaling van de concentratie van een bepaalde stof in een oplossing uit te voeren.
Men bepaalt in deze simulatie de zuur concentratie door er loog bij te druppelen.Een zuur bevat zure deeltjes,de zgn. waterstofionen. Deze zure deeltjes kan men met loog neutraliseren, en de oplossing wordt dan minder zuur. Dit drukt men uit in de zuurgraad of pH van de oplossing. Deze zuurgraad neemt waarden aan tussen 0 en 14. Als de zuurgraad 7.0 is dan noemt men de oplossing neutraal.

pH waarde schaal: N.B.: deze schaal is logaritmisch.

```
      1  2  3  4  5  7  8  9  10  11  12  13  14
  zeer zuur       neutraal                zeer basisch
```

Het punt waarbij de pH 7.0 wordt noemt men het omslagpunt of equivalentie punt. Indien de zuurgraad groter is dan 7.0 dan noemt men de oplossing basisch (dit is het tegenovergestelde van zuur). In de laboratorium opstelling gebruikt vaak een extra toegevoegde indicator stof. Deze indicator verandert van kleur op het omslagpunt. Dit is ook in de simulatieopstelling te zien.
Normaal stopt men dan op het omslagpunt met titreren en gaat men aan het rekenen om het gehalte aan zuur te bepalen.
In de simulatie druppelen we gewoon door om de hele grafiek te krijgen.
De standaardwaarden staan in de initialisatie procedure en kunnen naar wens aangepast worden.

In dit model gebruiken we:

Zoutzuur = HCl,een sterk zuur
Azijnzuur= HAc,een zwak zuur
Een combinatie van azijnzuur en zoutzuur in een grafiek.

Voor mensen met scheikunde kennis, is het verhaal misschien een te eenvoudige voorstelling van zaken, maar het principe wordt nu voor meer mensen duidelijk.
Zelf ben ik zeer geinteresseerd in simulaties op scheikundig, natuurkundig of biologisch gebied. Voor ideeen of programma's of verbeteringen van dit programma houd ik mij dan ook ten zeerste aanbevolen.

Gert Jan Noorland

```
 10 PROGRAM TITRATIE
 20 REM*** TITRATIE V1.0 ***
 30 REM **** AUTEUR : G.J.NOORLAND ****
 40 REM NODIG ZIJN P-CHARME + GAGS
 50 REM GOEDE SCHAKELSOFT BRANQUART.CK,ENZ
 60 REM GEHEUGEN VAN #2900-#6000
 70 REM IN DE INTIALISATIE KAN MEN HET SIMULATIE MODEL INSTELLEN
 80
 90 BASE#50;REM BASE ADRES SPRITES/PAINTS
100 DIM A(40),B(10)
110 FDIM %MM2,%VV2;GOS.z
120
130
140 REM*** main program ***
150 CLEAR 4
160 MENU
170 LINK#FFE3
180 GOTO 150
190 END
200
210 REM * PROC'S & CREATE'S *
220zPROC OPSTELLING,M,H
230  REM statief
240  CUBE 1,0,30,35,3,135
250  CUBE 1,0,30,31,45,3
260  REM buret
270  MOVE 39,170;DRAW 39,70;DRAW 50,65;DRAW 50,61
280  MOVE 52,61;DRAW 52,65;DRAW 43,71;DRAW 43,170
290  MOVE 47,67;DRAW 55,67;MOVE 55,70;DRAW 55,64
300  MOVE 34,75;DRAW 38,75;MOVE 34,77;DRAW 39,77
310  MOVE 40,165;DRAW 42,165
320  REM erlenmeijer
330  MOVE 46,61;DRAW 46,57;DRAW 38,38;DRAW 40,35;DRAW 60,35
340  DRAW 62,38;DRAW 54,57;DRAW 54,61;MOVE 41,42;DRAW 59,42
350  PAINT 50,41
360  PAINT 80,155,2
370  PAINT 10,176,2
380  PAINT 10,165,2
390  PAINT 250,125,2
400  PAINT 100,55,2
410  PAINT 41,90,4
420  PAINT 45,69,4
430  P=165;REM STARTSTAND BURET
440  MOVE 120,55
450 PEND
460
470 PROC WRITE(A,X,Y,S),I,J,B
480  J=0;B=S*LEN(A)-1
490  FOR I=X TO X+B STEP S
500    SET (A?J),I,(Y+6)
510    J=J+1
520  NEXT I
530 PEND
540
550 PROC MENU
560  CUBE 1,0,2,178,251,11
570  $A="TITRATIE";WRITE(A,99,180,7)
580  CUBE 1,0,2,178,83,11;PAINT3,179,7
590  CUBE 1,0,172,178,81,11;PAINT216,179,7
600  BORDER 0,0
```

```
 610  CUBE 1,0,3,1,249,29
 620  PAINT31,31,6;CUBE 1,0,50,130,136,10
 630  FOR I=1 TO 3
 640   CUBE 1,0,80,(130-I*15),90,10
 650   BLOCK 0,81,(131-I*15),88,9
 660  NEXT
 670  CUBE 1,0,50,(130-I*15),136,10
 680  BLOCK  0,51,131,133,9
 690  BLOCK  0,51,(131-I*15),133,9
 700  INV
 710  $A="Menu : ";WRITE(A,55,132,7)
 720  $A=" 1. sterk zuur";WRITE(A,82,117,6)
 730  $A=" 2. zwak zuur";WRITE(A,82,102,6)
 740  $A=" 3. twee zuren";WRITE(A,82,87,6)
 750  $A="keuzenummer: ?";WRITE(A,55,72,6)
 760  SET:COP,10,25;SET:ACB,225,25;SET:ACO,225,17
 770  $A="Noorland";WRITE(A,10,8,6)
 780  $A="simulatie";WRITE(A,100,15,7)
 790  DO  KEY C;UNTIL C>48 AND C<52
 800  INV
 810  FILL #81C0,#9800,0
 820  CASE C OF
 830<49> BUILDSCREEN;OPSTELLING;INV;STERK-ZUUR-PH
 840<50> BUILDSCREEN;OPSTELLING;INV;ZWAK-ZUUR-PH
 850<51> BUILDSCREEN;OPSTELLING;INV;STERK-ZUUR-PH;BLOCK 1,22,33,57,144;INV;OPSTELLING;INV;ZWAK-ZUUR-PH
 860 CEND
 870 PEND
 880
 890 PROC BUILDSCREEN
 900  BORDER1,1
 910  CUBE 1,0,3,1,249,29
 920  CUBE 1,0,20,31,60,147
 930  CUBE 1,0,8,2,110,10
 940  CUBE 1,0,160,2,80,11
 950  CUBE 1,0,8,18,110,10
 960  CUBE 1,0,160,18,80,11
 970  PAINT 5,3,6;PAINT 30,16,6;PAINT251,18,6;PAINT 4,28,6
 980  PAINT 251,7,6;PAINT 220,29,6;PAINT 140,10,6
 990  CUBE 1,0,120,55,98,110
1000  PAINT 150,60,3
1010  CUBE 1,0,102,38,130,136
1020  SET:VPIJL,106,169;SET:pH,112,172
1030  SET:HPIJL,170,49;SET:ml,182,47;SET:loog,200,47
1040  $A="0";WRITE(A,114,50,5)
1050  $A="1";WRITE(A,110,62,5)
1060  $A="3";WRITE(A,110,78,5)
1070  $A="5";WRITE(A,110,94,5)
1080  $A="7";WRITE(A,110,110,5)
1090  $A="9";WRITE(A,110,126,5)
1100  $A="11";WRITE(A,110,142,5)
1110  $A="13";WRITE(A,110,158,5)
1120  $A="20";WRITE(A,127,49,5)
1130  $A="40";WRITE(A,143,49,5)
1140  $A="60";WRITE(A,159,49,5)
1150  $A="80";WRITE(A,175,49,5)
1160  $A="100";WRITE(A,188,49,5)
1170  $A="120";WRITE(A,204,49,5)
1180  SET:loog,48,120
1190  SET:zuur,59,60
```

```
1200  $A="volume:";WRITE(A,12,5,6)
1210  SET:ml,90,10;SET:loog,100,10
1220  $A="pH :";WRITE(A,170,5,6)
1230 PEND
1240
1250 PROC FIX(%X,X,Y)
1260  C=2
1270  FIF %X<1;%X=0
1280  FIF %X>1;C=3
1290  FIF %X>=10;C=4
1300  FIF %X>=100;C=5
1310  STR %X,B
1320  B?C=#D
1330  BLOCK 1,(X-3),(Y-2),30,8
1340  WRITE(B,X,Y,5)
1350 PEND
1360
1370 PROC OMSLAG-KLEUR
1380  UNSET:DRUPPEL;PAINT 60,37,5
1390 PEND
1400
1410 PROC INITIALISATIE
1420  %D=1.0  ;REM DRUPPEL VOLUME
1430  %VV1=50 ;REM VOLUME ZUUR
1440  %VV2=0  ;REM TOEGEVOEGD LOOGVOLUME
1450  %MM1=0.1;REM MOLARITEIT ZUUR
1460  %MM2=0.1;REM MOLARITEIT LOOG
1470  %Z=(%VV1*%MM1)/%MM2-%D;REM EQUIVALENTIE PUNT
1480  %K=1.86E-4;REM ZUURCONSTANTE
1490  N=%(2/%D)
1500 PEND
1510
1520 PROC STERK-ZUUR-PH
1530  $A="NaOH + HCl";WRITE(A,30,20,6);$A="zuur: sterk";WRITE(A,170,20,6)
1540  INITIALISATIE
1550  WHILE %VV2< %Z
1560   %VV2=%VV2+%D
1570   %H=(%VV1*%MM1-%VV2*%MM2)/(%VV1+%VV2)
1580   FIF %H<=0;%P=7;GOTO 1600
1590   %P=-LOG(%H)/LOG(10)
1600   FIX(%VV2,60,5);FIX(%P,200,5);DRUP(P,Z);GRAFIC(%VV2,%P)
1610  WEND
1620  OMSLAG-KLEUR
1630  FOURTH-PART-PH
1640  P.$7$7$7$7
1650 PEND
1660
1670 PROC GRAFIC(%X,%Y)
1680  %F=.7;REM X-AS
1690  %G=.8;REM Y-AS
1700  FIF %(%V*%F) = %(Y*%G);GOTO 1730
1710  PLOT7,%(%X*%F+120),%(%Y*10*%G+57)
1720  %V=%Y
1730 PEND
1740
1750 PROC DRUP(:P,Z)
1760  SET:DRUPPEL,49,58
1770  FOR I=1 TO 15
1780  SHOVE:DRUPPEL,0,-1
1790  NEXT
```

```
1800  SOUND 150,10
1810  UNSET:DRUPPEL
1820  IF Z%N=0THEN BLOCK1,40,P,3,1;P=P-1
1830  Z=Z+1
1840 PEND
1850
1860 PROC ZWAK-ZUUR-PH
1870  BLOCK 1,9,19,79,9
1880  BLOCK 1,161,19,79,9
1890  $A="NaOH + HAc";WRITE(A,30,20,6);$A="zuur: zwak";WRITE(A,170,20,6)
1900  INITIALISATIE
1910  FIRST-PART-PH
1920  SECOND-PART-PH
1930  THIRD-PART-PH
1940  FOURTH-PART-PH
1950  UNSET:DRUPPEL;PAINT 60,37,5
1960  P.$7$7$7$7
1970 PEND
1980 PROC FIRST-PART-PH
1990  %H=(-%K+SQR((%K*%K)+4*%K*%MM1))/2
2000  %P=-LOG(%H)/2.302
2010  MOVE(120+%(%VV2)),(55+%(%P))
2020   FIX(%VV2,60,5);FIX(%P,200,5);DRUP(P,Z);GRAFIC(%VV2,%P)
2030 PEND
2040
2050 PROC SECOND-PART-PH
2060  WHILE %VV2<%Z
2070   %VV2=%VV2+%D
2080   %H=(%K*(%VV1*%MM1-%VV2*%MM2)/(%VV1+%VV2))/((%VV2*%MM2)/(%VV1+%VV2))
2090   %P=-LOG(%H)/2.302
2100   FIX(%VV2,60,5);FIX(%P,200,5);DRUP(P,Z);GRAFIC(%VV2,%P)
2110  WEND
2120 PEND
2130
2140 PROC THIRD-PART-PH
2150  %P=+0.5*14-0.5*LOG(%K)/2.302+0.5*LOG((%VV1*%MM1)/(%VV1+%VV2))/2.302
2160   FIX(%VV2,60,5);FIX(%P,200,5);DRUP(P,Z);GRAFIC(%VV2,%P)
2170  OMSLAG-KLEUR
2180 PEND
2190
2200 PROC FOURTH-PART-PH
2210  WHILE %VV2 < 2*%VV1
2220   %VV2=%VV2+%D
2230   %H=(%VV2*%MM2-%VV1*%MM1)/(%VV1+%VV2)
2240   FIF %H <=0;GOTO 2270
2250   %O=-LOG(%H)/2.302
2260   %P=14-%O
2270   FIX(%VV2,60,5);FIX(%P,200,5);DRUP(P,Z);GRAFIC(%VV2,%P)
2280  WEND
2290 PEND
2300
2310 CREATE/P:1 #A,#A,#A,#A,#A,#A,#A,#A
2320 CREATE/P:2 #44,#11,#44,#11,#44,#11,#44,#11
2330 CREATE/P:3 #10,0,#10,0,#10,0,#10,#AA
2340 CREATE/P:4 #AA,#66,#AA,#66,#AA,#66,#AA,#66
2350 CREATE/P:5 #92,0,#49,0,#24,0,#59,0
2360 CREATE/P:6 #55,#55,#55,#55,#55,#55,#55,#55
2370 CREATE/P:7 #FF,0,#FF,0,#FF,0,#FF,0
2380
```

```
2390 CREATE A,0,#88,#88,#88,#F8,#88,#88,#70,0,0,0,0,0,0,0/A:CH"A"
2400 CREATE C,0,#70,#88,#80,#80,#80,#88,#70,0,0,0,0,0,0,0/A:CH"C"
2410 CREATE E,0,#F8,#80,#80,#F0,#80,#80,#F8,0,0,0,0,0,0,0/A:CH"E"
2420 CREATE H,0,#88,#88,#88,#F8,#88,#88,#88,0,0,0,0,0,0,0/A:CH"H"
2430 CREATE I,0,#70,#20,#20,#20,#20,#20,#70,0,0,0,0,0,0,0/A:CH"I"
2440 CREATE M,0,#88,#88,#88,#A8,#A8,#D8,#88,0,0,0,0,0,0,0/A:CH"M"
2450 CREATE O,0,#70,#88,#88,#88,#88,#88,#70,0,0,0,0,0,0,0/A:CH"O"
2460 CREATE N,0,#88,#88,#88,#98,#A8,#C8,#88,0,0,0,0,0,0,0/A:CH"N"
2470 CREATE R,0,#88,#90,#A0,#F0,#88,#88,#F0,0,0,0,0,0,0,0/A:CH"R"
2480 CREATE T,0,#20,#20,#20,#20,#20,#20,#F8,0,0,0,0,0,0,0/A:CH"T"
2490 CREATE SPATIE,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0/A:CH" "
2500 CREATE ?,0,#20,0,#20,#10,8,#88,#70,0,0,0,0,0,0,0/A:CH"?"
2510 CREATE MIN,0,0,#F8,#F8,0,0,0,0,0,0,0,0,0,0,0/A:CH"-"
2520 CREATE COLON ,0,#60,#60,0,#60,#60,0,0,0,0,0,0,0,0,0/A:CH":"
2530 CREATE PLUS,0,0,#20,#20,#F8,#20,#20,0,0,0,0,0,0,0,0/A:CH"+"
2540 CREATE DRUPPEL ,0,0,0,0,#60,#F0,#F0,#60,0,0,0,0,0,0,0
2550 CREATE ACB,0,#F,#F,#F,#F,7,7,3,0,#F0,#F0,#F0,#F0,#E0,#E0,#C0
2560 CREATE ACO,0,1,7,#19,3,#F,#1F,#1F,#7C,#ED,#80,#80,#C0,#F0,#F8,#F8
2570 CREATE COP,#1E,#21,#5E,#50,#50,#5E,#21,#1E,0,0,#80,#80,#80,#80,0,0
2580 CREATE ONE  ,0,#20,#20,#20,#60,#20,0,0,0,0,0,0,0,0,0/A:CH"1"
2590 CREATE TWO  ,0,#F0,#40,#20,#90,#60,0,0,0,0,0,0,0,0,0/A:CH"2"
2600 CREATE THREE,0,#F0,#10,#20,#90,#70,0,0,0,0,0,0,0,0,0/A:CH"3"
2610 CREATE FOUR ,0,#20,#F0,#A0,#40,#20,0,0,0,0,0,0,0,0,0/A:CH"4"
2620 CREATE FIVE ,0,#F0,#10,#F0,#80,#F0,0,0,0,0,0,0,0,0,0/A:CH"5"
2630 CREATE SIX  ,0,#60,#90,#E0,#40,#20,0,0,0,0,0,0,0,0,0/A:CH"6"
2640 CREATE SEVEN,0,#80,#40,#20,#10,#F0,0,0,0,0,0,0,0,0,0/A:CH"7"
2650 CREATE EIGHT,0,#60,#90,#60,#90,#60,0,0,0,0,0,0,0,0,0/A:CH"8"
2660 CREATE NINE ,0,#60,#10,#70,#90,#60,0,0,0,0 0,0,0,0,0,0/A:CH"9"
2670 CREATE ZERO ,0,#40,#A0,#A0,#A0,#40,0,0,0,0 0,0,0,0,0,0/A:CH"0"
2680 CREATE POINT,0,#40,0,0,0,0,0,0,0,0,0,0,0,0,0/A:CH"."
2690 CREATE pH,#80,#80,#E4,#94,#97,#E4,4,4,0,0,#80,#80,#80,#80,#80,#80
2700 CREATE ml,0,0,#8A,#AA,#DA,2,2,2,0,0,0,0,0,0,0
2710 CREATE loog,0,0,#98,#A5,#A5,#98,#80,#80,6,1,#C3,#25,#25,#C3,0,0
2720 CREATE zuur,0,0,#F7,#45,#25,#F5,0,0,0,0,#74,#54,#56,#55,0,0
2730 CREATE VPIJL,#40,#40,#40,#40,#40,#40,#E0,#40,0,0,0,0,0,0,0,0
2740 CREATE HPIJL,#00,#02,#FF,#02,#00,0,0,0,0,0,0,0,0,0,0,0
2750 CREATE a,0,#74,#88,#78,8,#70,0,0,0,0,0,0,0,0,0/A:CH"a"
2760 CREATE c,0,#78,#80,#80,#80,#78,0,0,0,0,0,0,0,0,0/A:CH"c"
2770 CREATE d,0,#70,#90,#90,#90,#70,#10,#10,0,0,0,0,0,0,0/A:CH"d"
2780 CREATE e,0,#78,#80,#F8,#88,#70,0,0,0,0,0,0,0,0,0/A:CH"e"
2790 CREATE i,0,#20,#20,#20,#20,0,#20,0,0,0,0,0,0,0,0/A:CH"i"
2800 CREATE k,0,#48,#70,#50,#48,#40,#40,#40,0,0,0,0,0,0,0/A:CH"k"
2810 CREATE l,0,#70,#20,#20,#20,#20,#20,#60,0,0,0,0,0,0,0/A:CH"l"
2820 CREATE m,0,#A8,#A8,#A8,#A8,#70,0,0,0,0,0,0,0,0,0/A:CH"m"
2830 CREATE o,0,#60,#90,#90,#90,#60,0,0,0,0,0,0,0,0,0/A:CH"o"
2840 CREATE n,0,#90,#90,#90,#90,#E0,0,0,0,0,0,0,0,0,0/A:CH"n"
2850 CREATE p,#80,#80,#E0,#90,#90,#E0,0,0,0,0,0,0,0,0,0/A:CH"p"
2860 CREATE r,0,#80,#80,#80,#C0,#B0,0,0,0,0,0,0,0,0,0/A:CH"r"
2870 CREATE s,0,#F0,8,#70,#80,#78,0,0,0,0,0,0,0,0,0/A:CH"s"
2880 CREATE t,0,#30,#40,#40,#E0,#40,#40,0,0,0,0,0,0,0,0/A:CH"t"
2890 CREATE u,0,#68,#90,#90,#90,#90,0,0,0,0,0,0,0,0,0/A:CH"u"
2900 CREATE v,0,#20,#50,#88,#88,#88,0,0,0,0,0,0,0,0,0/A:CH"v"
2910 CREATE w,0,#50,#A8,#A8,#88,#88,0,0,0,0,0,0,0,0,0/A:CH"w"
2920 CREATE z,0,#F8,#40,#20,#10,#F8,0,0,0,0,0,0,0,0,0/A:CH"z"
2930 RETURN
```

=====================
***** BLACK JACK *****
=====================

Toen ik in juli van dit jaar met mijn gezin in de USA was, hebben wij ondermeer een bezoek gebacht aan Las Vegas, de beroemde gokstad. Deze stad bestaat alleen uit casino's, wedding-schapels en hotels/motels. Tijdens dit bezoek hebben wij uiteraard ook enige gokhallen bezocht en een klein gokje gewaagd aan een 'slot' (een eenarmige bandiet) en aan de black-jack tafel (een aangepaste versie van ons een en twintigen).

Van dit BLACK-JACK heb ik nu een computer-versie gemaakt voor de ons atoompje. Na het opstarten van het programma wijst het spel zich zelf, de regels zijn hetzelfde als bij Een en Twintigen. Alleen blijkt uit dit spel waarom de bank (bijna) altijd wint.

Ik hoop niet dat u aan dit spel verslaaft zult raken, zoals menigeen in Las Vegas aan het echte spel wel is.

DRAWPOKER

Drawpoker is een gokspel dat veel wordt gespeeld in Las Vegas. De bedoeling is dat u een, twee, drie, vier of vijf munten (meestal quarters, 25 dollarcent) inzet. U begint het spel door de DRAW-toets in te drukken. U krijgt vijf kaarten met de rug naar u toe te zien. Deze worden een voor een omgedraaid. Als ze allemaal zijn omgedraaid moet u beslissen welke kaarten u laat staan en welke u wilt vervangen door een andere kaart. Dit doet u door de HOLD-toets die hoort bij een van de kaarten die u wilt laten staan in te drukken. Als u dat gedaan heeft ziet u onder deze kaart het woord HOLD (of HELD) verschijnen. Drukt u een tweede keer op dezelfde HOLD-toets dan verdwijnt dat woord weer. Heeft u alle kaarten die u wilt laten staan gemerkt met het woord HOLD (of HELD) dan drukt u de DRAW-toets in. Op de niet-gemerkte kaarten worden andere gelegd en omgedraaid. U kunt natuurlijk ook geen enkele kaart laten staan door meteen op de DRAW-toets te drukken. Nu kijkt het gokmachien of uw kaarten winnende kaarten zijn. Zo nee, dan bent u uw geld kwijt. Zo ja,dan krijgt u een aantal munten uitgekeerd. Dit aantal is minimaal gelijk aan uw inzet.

Deze Drawpoker werkt bijna hetzelfde. Omdat onze Atom niet is uitgerust met een inwerpgleuf voor kwartjes (het meine in ieder geval niet), heb ik het inzetten iets anders moeten maken. Als u het programma start ziet u een paar rechthoekjes en vierkantjes op het scherm verschijnen. Als het programma stopt met tekenen dan is de tijd gekomen om uw inzet te bepalen. Dit doet u door zoveel keren als u munten in wilt zetten op een (alfanumerieke) toets te drukken. Als u op de spatiebalk drukt of u hebt de maximum inzet van vijf munten bereikt, dan begint het spel (zie hoofdstukje hiervoor). De HOLD-toetsen zijn de toetsen 1,2,3,4 en

5. De DRAW-toets is de spatiebalk.
U heeft een beginkapitaal van 20 munten en 0 punten. Elke keer als u iets wint wordt dat bij het aantal munten en uw punten opgeteld. Als u munten inzet wordt dat niet van uw punten afgetrokken. Als u wint ziet u het woord WIN knipperen, als u niet wint ziet u SORRY staan. Nu moet u weer inzetten. Wilt hetzelfde inzetten als de vorige keer dan drukt u op de spatiebalk. Zo niet dan drukt u weer een aantal keren op een andere toets dan de spatiebalk. Als u geen munten meer heeft dan is uw uiteindelijke score in de rechterbovenhoek te zien en het spel is afgelopen. In de linkerbovenhoek ziet u het aantal munten dat u nog heeft, tussen dit en de score ziet u hoeveel munten u heeft ingezet en daaronder het aantal munten dat u wint.

Het aantal munten dat u kunt winen is afhankelijk van uw inzet en uw kaarten.

Een paar:
Twee boeren (J), twee vrouwen (Q), twee heren (H) of twee azen (A).

Twee paar
Drie dezelfde
Straat:
Opeenvolgende kaarten, hoeven niet dezelfde kleur te hebben.
Grote straat:
Straat van 10 tot en met AAS
Flush:Alle kaarten met dezelfde kleur.
Full house:
Een paar en drie dezelfde kaarten
Vier dezelfde kaarten
Straat-Flush:
Straat in een kleur.
Grote straat-Flush
Grote straat in een kleur.

Etienne Thijsse
Albert Verweystraat 20
Alphen a/d rijn

B I G B E N N Y

(of: m'n computer is bij de tijd)

Na het inbouwen van big benny krijgen we, wanneer de computer aan staat, de beschikking over de momentele tijd, de datum, en de weekdag, die we met behulp van een statement (onder P-charme, of SOS) of een LINK naar een bepaald adres kunnen oproepen.
Het grote voordeel van big benny ligt hierin, dat de tijd doorloopt, wanneer de computer uit staat. Er is nl. een mogelijkheid om een batterijtje (of accu) aan te sluiten. Dus: battery back-up: altyd de juiste tyd, datum, en dag beschikbaar (zolang er nog spanning over de battery staat...., en dat is erg lang)

Hoe gaat het praktisch ??

Zoals al gezegd is, is big benny een klein printje, met een klok IC, en een PIA (een I/O IC, vergelijkbaar met de VIA).
Het klok IC is van het type MSM 5832, en wordt door de club bij het printje geleverd. De PIA, en wat klein materiaal, is in de meeste electronica zaken verkrijgbaar. Het in elkaar solderen van het geheel is niet erg moeilijk. Wanneer u het toch niet aandurft is er vast in uw regio wel iemand die beter 'soldeervaardig' is.
Wanneer het geheel in elkaar zit, kan Big Benny z'n plaatsje op PL8 (Deze connector zit aan de soldeerzijde van de computerprint, rechts boven het toetesenbord.) krijgen. De soldeerzijde van Big Benny naar boven

Hoe laat is het ??

Een gezegde uit de computerwereld luidt "HARDware kan NIETS zonder SOFTware". Dit is ook van toepassing op onze Big Benny. Er zal dus ergens een klein programmatje moeten komen, dat de klok kan uitlezen, en gelijkzetten.
Het is erg makkelijk om dit programma'tje op de geheugenkaart, of schakelkaart, onder Battery Back Up te zetten, zodat dit altijd beschikbaar is. Bij het gebruik van P-charme of een SOS kunnen voor de besturing van Big Benny een aantal statements maken (TIME, DATE, DAY). Bij een systeem zonder SOS of P-CHARME [wie heeft dit eigenlijk nog?] kan het programmatje 'aan geLINKed' worden, waarna de tijd, datum, of dag in de string buffer komt te staan.

Software

De software is reeds eerder in (toen) Acorn Nieuws verschenen, en wel in no.4 van 1983. Het lijkt mij zinvol deze software nogmaals te publiceren, en dan in een versie als statement, en in een versie voor de systemen zonder P-CHARME, of SOS.

Hoe moet het 1 ? (bij statements)

Heel makkelijk. Wanneer de statements TIME, DATE, en DAY door het systeem herkend worden,

tijd, datum, of dag zetten: bijv.
TIME="09:30", DATE="16-09-85", DAY="1" (Day=0: zondag)

Tijd, datum, of dag lezen: bijv.
TIME$A : zet de tijd in $A
TIME : drukt de tijd af (gaat verder als print statement)
TIME" : drukt tijd af, en slaat 2 regels over.

Hoe moet het 2 ? (bij losse routine)

Om de klok gelijk te zetten (schrijven dus) moet de accumulator gelijk zijn aan #40, om te lezen moet A geijk zijn aan #60.
Verder moet voor de tyd het X register #50 zijn, voor de datum: x=#80, en voor de dag x=#60.
De informatie komt (bij lezen) terecht in de stringbuffer op #140, om te schrijven moet de deata ook op #140 staan.
Een voorbeeld:
Gelijkzetten op 11:00, maandag 16 sept 1985
A=#40;X=#50;$#140="11:00";LINK #yyyy
A=#40;X=#80;$#140="16-09-85";LINK #yyyy
A=#40;X=#60;$#140="1";LINK #yyyy
waarbij yyyy staat voor het startadres van de routine.

De tijd lezen:

A=#80;x=#50;LINK #yyyy;P.$#140
en de tijd verschijnt!.

Verdere mogelijkheden:

Er is ook een mogelijkheid om Big Benny een interrupt te laten genereren (iedere seconde, minuut, of uur). Hiermee is ook heel wat interessants te doen (bijv. een wekker programma) Zie hiervoor Acorn nieuws no. 1, 1984; pag.86

Op het printje zijn nog 6 PIA I/O poorten over. Hiermee kunnen verschillende dingen geschakeld worden (bijv. de minischakelkaart, of andere bank-switchingen in de computer)

Al met al is Big Benny dus een leuke uitbreiding van de computer. Je blijft er mee bij de tijd!!
Succes met het bouwen! Ten overvloedde vermeld ik hier nog even de componentenwaaden (De nummers refereren aan de printopdruk) [de complete bouwbeschrijving is te vinden op pag. 45 van Acorn Nieuws Augustus 1983.]

R 1...14	: 10 K	
R 15	: 5 K	
R 16	: 560 ... 1K2	(afhankeijk v.d. laadstroom)
R 17	: 51 K	
C 1	: Trimmer 22 pF	
C 2	: 5p6	(Zie opmerking)
C 3,4	: 1 uF	(tantaal elco 10 V)
T 1,2	: BC 547	
T 3	: BC 557	
D	: AA 119	
IC 1	: 6821 (PIA)	
IC 2	: MSM 5832 RS	
X	: Kristal 32,768 KHz (uit bijv. oud kwarts horloge)	

OPMERKING : Wanneer de klok niet gelijklopend te krijgen is, dan C2 (5p6) vervangen door 22pF, dan is het regelbereik van de trimmer groter, en zal de klok WEL gelijklopend te krijgen zijn.

Rudi van Drunen.

```
 10 PROGRAM TIME
 20 REM big benny CONTROL
 50
 60 DIM LL15;F.I=0 TO 15;LLI=-1;N.
 75 P.$12;IN. "ROUTINE OP" A
 80 PRINT $21;P=A;GOSUB b
 90 PRINT $6 ;P=A;GOSUB b
 95 @=0;P."AANROEPEN MET LINK #" A
170 END
180
190bB=#B400
200[
210 STA#80;STX#81
220:LL2 LDA@#40;STA#52;LDA@#01;STA#53
230:LL3 LDA B+1;PHA;LDA B+3;PHA;LDY@#00
240 STY B+3;LDA B+2;ORA@#C0;STA B+2;LDX@#04
250 STX B+3;LDA B+2;AND@#3F;STA B+2;PHA
260 LDA@#34;STA B+3;STY B+1;LDA@#FF;BIT#80
270 BPL LL4;LDA@#F0
280:LL4 STA B;STX B+1
290:LL5 LDA(#52),Y;AND@#0F;ORA#81;LDX#81;CPX@#50
300 BNELL6;ORA@#08
310:LL6 STA B;PLA;PHA;ORA#80;STA B+2
320 LDA B;AND@#0F;ORA@#30;CPX@#80;BEQ LL7
330 CPX@#50;BNE LL8
340:LL7 AND@#F3
350:LL8 STA(#52),Y;PLA;PHA;STA B+2;SEC
360 TXA;SBC@#10;STA#81;INY;LDX@#3A
370 CMP@#50;BEQLL12;BCCLL9;ADC@#3F;LDX@#2D
380:LL9CPY@#02;BEQLL10;CPY@#05;BNELL11
390:LL10STA#81;TXA;STA(#52),Y;INY
400:LL11CPY@#08;BNELL5
410:LL12LDA@#FF;STA B;PLA;ORA@#80;STA B+2
420 PLA;ORA@#38;STA B+3;PLA;STA B+1
430 LDA@#0D;STA(#52),Y;RTS
510];R.
```

Literatuur:

1. Datasheet MSM5832 (Drukwerkarchief, regio ??)
2. Datasheet 6821 (Drukwerkarchief, regio ??)
3. ACORN NIEUWS jrg.2, No.4 [Aug. 1983]
4. ACORN NIEUWS jrg.3, No.1 [Maart 1984]

```
 10 PROGRAM TIME
 20 REM TIME,DATE,DAY
 30 REM STATEMENTS FOR USE
 40 REM WITH p-charme OR sos
 50
 60 DIM LL15;F.I=0 TO 15;LLI=-1;N.
 70 DIM DD(1);DD0=-1
 80 PRINT $21;P=A;GOSUB b
 90 PRINT $6 ;P=A;GOSUB b
100 $T="TIME";T=T+LEN(T)
110 ?T=DD0/256!#80;T?1=DD0%256;T=T+2
120 $T="DATE";T=T+LEN(T)
130 ?T=(DD0+3)/256!#80;T?1=(DD0+3)%256;T=T+2
140 $T="DAY";T=T+LEN(T)
150 ?T=(DD0+6)/256!#80;T?1=(DD0+6)%256;T?2=#80;T=T+2
160 A=P;T!1=A
170 END
180
190bB=#B400
200[
220:LL2 LDA@#40;STA#52;LDA@#01;STA#53
230:LL3 LDA B+1;PHA;LDA B+3;PHA;LDY@#00
240 STY B+3;LDA B+2;ORA@#C0;STA B+2;LDX@#04
250 STX B+3;LDA B+2;AND@#3F;STA B+2;PHA
260 LDA@#34;STA B+3;STY B+1;LDA@#FF;BIT#80
270 BPL LL4;LDA@#F0
280:LL4 STA B;STX B+1
290:LL5 LDA(#52),Y;AND@#0F;ORA#81;LDX#81;CPX@#50
300 BNELL6;ORA@#08
310:LL6 STA B;PLA;PHA;ORA#80;STA B+2
320 LDA B;AND@#0F;ORA@#30;CPX@#80;BEQ LL7
330 CPX@#50;BNE LL8
340:LL7 AND@#F3
350:LL8 STA(#52),Y;PLA;PHA;STA B+2;SEC
360 TXA;SBC@#10;STA#81;INY;LDX@#3A
370 CMP@#50;BEQLL12;BCCLL9;ADC@#3F;LDX@#2D
380:LL9CPY@#02;BEQLL10;CPY@#05;BNELL11
390:LL10STA#81;TXA;STA(#52),Y;INY
400:LL11CPY@#08;BNELL5
410:LL12LDA@#FF;STA B;PLA;ORA@#80;STA B+2
420 PLA;ORA@#38;STA B+3;PLA;STA B+1
430 LDA@#0D;STA(#52),Y;RTS
440:DD0 LDX@#50;BIT #80A2
450 BIT#60A2;STX#81;LDA@#80;STA#80;JSR #F291
460 CMP@#3D;BEQ LL15;CMP@#24;BNE LL14;JSR #C78B
470 JSR #C3CB
480:LL13 JSR #C4E4;JSR LL3;JMP #C55B
490:LL14 DEC#03;JSR LL2;JMP #C39C
500:LL15 JSR #CEB1;LSR#80;BNE LL13
510];R.
```

SALFAA QUICK REFERENCE CARD (g.j.noorland)

COMMANDO'S:

ASM-<beginletter B C I S U V >
PASS <cijfer 0 1 2 >
CALL <adres> werkt als LINK in BASIC.
(<letter> rest commentaar tot ')'
\ commentaar tot CR of ';'

ASM-B : symbolentabel op #8200 variabele S
objectcode pointer #9000 variabele P
geheugenpointer #0000 variabele O
nieuwe breakhandler gezet,oude bewaard
ASM-C : nieuwe breakhandler gezet,oude bewaard.
ASM-I : zie ASM-B,nu scherm uit
ASM-S : geeft de symbolentabel, ook .S
ASM-U : sorteert de symbolentabel, ook .U
ASM-V : SALFAA versie no. afgedrukt, ook .V

DIRECTIVES

(max.26,elke letter eenmaal)

.ASCII\string\,byte1,byte2,enz. Deze worden achtereenvolgens gepoked.
.BYTE leest een of meer expressies,plaatst laagste bytes ervan in geheugen, ook strings.
.CODE leest adres en plaatst dit in P, assembleer adres.
.DBYTE plaatst 2 bytes adres in geheugen, eerst highbyte dan lowbyte.
.END printer uit, scherm aan, naar BASIC-mode.
.FIRST forceert listing bij PASS 0.
.HARD tijdens PASS 1 de printer aan, eventueel <SHIFT> voor vervolg listing.
.IF te vergelijken met IF in BASIC, maar zonder THEN.
.LIST normaal beide PASSes niet gelist, nu bij PASS 1 scherm aan.
.NEWLINE gevolgd door 1 byte expressie, geeft evenzo vele linefeeds in de listing.
.OPTION beinvloed 6 hoogste bits van de PROCESSORSTATUS byte.
.PRINT leest een of meer expressies. Tijdens PASS 1 naar printer, scherm uit.
.RAM leest adres in en plaatst dit in variabele O.
.SYMBOL geeft symbolentabel. Stop en start met <REPT>.
.TABLE leest beginadres en/of eindadres van de tabel in.
.UNRAVEL sorteert de symbolentabel.
.VERSION drukt SALFAA versie nummer af.
.WORD plaatst twee bytes adres in geheugen, low byte eerst.
.X reset niet beginadres tabel, maar initieert deze juist.

SYMBOLEN:

Definieert men met: dubbele punt<naam> ,dus :SYMBOOLNAAM
kleine en hoofdletters toegestaan. Max. 26 letters + de quote(').

DOS EN COS COMMANDOS: (alleen de * commando's)

PROCESSORSTATUS BYTE:

BIT	0	1	
0	eerste	laatste	PASS
1	wel	niet	wegschrijven
2	xxx	xxx	gereserveerd
3	-	-	--------
4	64+	32	karakters
5	geen	wel	pagecrossing melding
6	wel	geen	P-CHARME aanwezig
7	-	-	--------

N.B. : bij immedidiate adressing mag men binair invoeren, waarbij spaties niet tellen.

GAGS QUICK REFERENCE CARD

(g.j.noorland)

INPUT STATEMENTS:

```
JOYSTK A,B,C
AT KEY (A,B,C,.....)(a,b,c,.....)
```

GRAPHIC DESIGN SUPPORT:

```
INV
BORDER x,y
PAINT x,y,p                        p=nummer kleurpatroon
CREATE/P:p a1,a2,a3,a4,a5,a6,a7,a8
CUBE p,m,x,y,b,h                   p zie * en m zie **
CIRCLE p,x,y,r                     p zie *
PIXEL a,b,z                        z=1 pixel is set en z=0 unset
WINDOW m,x,y,b,h                   m=1 binnen en m=2 buiten raam tekenen
WOFF
FILL a,b,x                         x is waarde gevuld in adres a tot b
SNOW en NOSNOW
SCROLL U(p); SCROLL D(own); SCROLL R(ight); SCROLL L(eft)
HLINE p,x1,y1,x2,y2,l,f,A,B        p zie *; l x-coordinaat eindpixel komt in A
VLINE p,x1,y1,x2,y2,l,f,A,B        in B komt y-coordinaat eindpixel
PAUSE a                            pauze van 1/60 seconde
INK a
PAPER a
MODE a
BLOCK m,x,y,b,h                    m zie **
SOUND p,d                          p=pitch 1<=p<=255; d=duur
```

GAMES SUPPORT:

```
CREATE <name>,a1,a2,...........,a16
CREATE <name>,a1,a2,...........,a16
DEF <name>,0000000000000000
DEF:       1111000000001111
nog 6 maal
BASE q                             q=high byte van opslag create's
ASSIGN <name>,a                    nummer a bij sprite <name>
DEASS <name>
KILL <name>          of  KILL a
SET:<name>,x,y       of  SET a,x,y
UNSET:<name>         of  UNSET a
IMAGE:<name>,x,y     of  IMAGE a,x,y
TURN:<name>          of  TURN a
CARRY:<name>,x,y     of  CARRY a,x,y
SHOVE:<name>,dx,dy   of  SHOVE a,dx,dy
POS:<name>,X,Y       of  POS a,X,Y
ATHIT:<name1>,;<name2>;  of  ATHIT:<name1>,b;
ATHIT a,<name2>;         of  ATHIT a,b;
```

```
**  m=0 tekent altijd                              *  p=0  reset pixel
    m=1 tekent tot pixel set                          p=1  set pixel
    m=2 begint als pixel geset is                     p=2  invert pixel
    m=3 tekent tot en start weer bij e.v. set pixel
    m=4 begint bij set pixel en stopt bij volgende
```

PRCODE statement

Met behulp van dit statement kan ieder soort data naar de printer gestuurt worden, zelfs 2,3 en code (?#FE).
Gebruik: PRCODE 27,45,1 of PRCODE 27,3,2,10 of code=18;PRCODE 27,code enz.
Enkele bijzonderheden van het programma:
*Er wordt gekeken of de printer al aan stond, zoniet dan wordt na de verzonde data de printer weer uitgezet, zoja dan blijft de printer aan.
*De BRK-vector wordt verzet: evt.foutmeldingen worden niet naar de printer verzonden en de al verzonden codes worden zo veel moglijk weggehaald.
*Let op het JMP adres naar de PCHARME, deze maakt het mogelijk ook exvar variabelen te gebruiken.

Veel plezier met uw printer,
Marcel Wagenaar

```
 10 PROGRAM STAT.PRCODE
 20 PRINT $21;P=A;GOSUB a
 30 P=A;GOSUB a;PRINT $06
 40 $T="PRCODE";T=T+L.T
 50 T?0=start/256|#80
 60 T?1=start%256
 70 T?2=#80;T=T+2
 80 A=P;T!1=A
 90 END
100
110a[
120.TA #9400
130:STAT          =#05
140:STATINDEX     =#03
150:PCR           =#B80C
160:TESTBYTE      =#60
170:STAPEL        =#61
180:KEER          =#63
190:BRKVEC        =#202
200
210:SPATIE        =#20
220:KOMMA         =#2C
230:PUNTKOM       =#3B
240:CR            =#0D
250
260:BASIC         =#C558
270:DRUKAF        =#FEFB
280:HAALDATA      =#C78B
290:SPRINGKOMMA   =#C231
300:PRINT         =#FF0B
310:PCHARME       =#ABD5
320
330
340                                  \VERZET BRK-VECTOR:FOUT WORDT
350                                  \NIET OP DE PRINTER AFGEDRUKT
360:start   JSR init                 \initialiseer controlebyte 's
370                                  \TEST OF DE PRINTER
```

```
380                             \STANBY-/OF AAN STAAT
390          LDA PCR            \laadt controle byte
400          AND @#0E           \uitkomst nul? (=printer uit)
410          BNE kring          \nee,vervolg met eigenlijke rout.
420          LDA @#02           \ja !.zet hem aan
430          STA TESTBYTE       \set  testbyte
440          JSR DRUKAF         \met accu=2 gelijk aan print $2
450
460                             \EIGENLIJKE ROUTINE
470:kring    JSR blanco         \negeer spatie's
480          STY STATINDEX      \bewaar positie
490          CMP @KOMMA         \is het een komma ?
500          BNE puntkom        \nee,test voor puntkomma
510          INY                \y=y+1
520          BNE kring          \vervolg zoeken
530:puntkom CMP @PUNTKOM        \is het een puntkomma ?
540          BEQ terug          \ja,terug naar basic
550          CMP @CR            \is het einde van de regel ?
560          BEQ terug          \ja,terug naar basic
570          JSR HAALDATA       \haal de waarde op
580          LDX @#00           \reset stack index
590          STX #04            \is de
600          LDA #25            \waarde
610          ORA #34            \groter
620          ORA #43            \dan 255 ?
630          BNE fout           \ja,foutmelding
640          LDA #16            \laadt waarde
650          INC KEER           \aantal=aantal+1
660          JSR drukaf         \stuur hem naar printer
670          JMP kring          \begin opnieuw
680
690:terug    JSR herbrk         \herstel brk-vector
700          LDA TESTBYTE       \laadt controlebyte
710          CMP @#02           \stond printer aan ?
720          BNE basic          \ja,spring naar basic
730          LDA @#03           \
740          JSR DRUKAF         \zet de printer uit
750:basic    JMP BASIC
760
770
780                             \AFDRUK ROUTINE
790:drukaf   PHA
800          JMP PRINT
810
820                             \NUMMER TE GROOT
830                             \voor synchronisatie:
840                             \verwijder al verzonden code's
850:fout     LDX KEER           \laadt het aantal
860          LDA @#7F           \laadt delete karakter
870:delete   JSR DRUKAF         \stuur hem naar printer
880          DEX                \alle code's uitgewist ?
890          BNE delete         \nee herhaal
900          LDA @109           \="number too large"
910          PHA
920          PHA
930
940                             \FOUT OPVANG ROUTINE
950:brk      PLA                \zet stack
960          PLA                \goed
```

```
 970          CMP @29          \is het misschien een muvar ?
 980          BNE vervolg      \nee,ga door met normale afwerking
 990          PHA              \ga maar
1000          PHA              \kijken of het
1010          JMP PCHARME      \een muvar is
1020:vervolg  PHA              \foutnummer
1030          PHA              \op stack
1040          LDA @#03         \zet eerst
1050          JSR DRUKAF       \de printer uit
1060          JSR herbrk       \herstel brk-vector
1070          JMP (BRKVEC)     \spring naar oude brk-handler
1080
1090                           \NEGEER SPATIE'S
1100:herhaal  INY
1110:blanco   LDA (STAT),Y     \laadt statement plaats
1120          CMP @SPATIE      \is het een spatie ?
1130          BEQ herhaal      \ja,zoek volgende spatie
1140          RTS
1150
1160:init     LDA @#00         \reset
1170          STA TESTBYTE     \controlebyte
1180          STA KEER         \en aantal,
1190          LDA BRKVEC       \zet
1200          STA STAPEL       \oude
1210          LDA BRKVEC+1     \brk-vector
1220          STA STAPEL+1     \weg,
1230          LDA @brk%256     \installeer
1240          STA BRKVEC       \de
1250          LDA @brk/256     \nieuwe
1260          STA BRKVEC+1     \brk-vector
1270          RTS
1280
1290                           \HERSTEL BRK-VECTOR ROUTINE
1300
1310:herbrk   LDA STAPEL
1320          STA BRKVEC
1330          LDA STAPEL+1
1340          STA BRKVEC+1
1350          RTS
1360]
1370 RETURN
```

De LINE funktie

De oorsprong van deze funktie komt uit de COMBOX van Bram Poot. Mijn dank hiervoor. Ik heb het statement omgebouwd tot een funktie. Met funkties kun je immers meer doen.

Deze funktie moet als volgt gebruikt worden:

PRINT LINE 120 drukt de waarde van de geheugenplaats af waar regel 120 begint.

A=120

B=LINE A B neemt de waarde aan van de geheugenplaats waar regel 120 begint.

Dit laatste is handig bij bijvoorbeeld het HEX statement, helaas pakt de PCHARME HEX LINE 120 niet, hiervoor moeten we hem wijzigen (A=120;HEX A gaat wel).

Het programma gebruikt MINIAS bij het uitwerken van de assembler, eventueel de namen omwerken tot gewone labels.

Veel succes.
Marcel Wagenaar

```
 10 PROGRAM LINE-FUN
 20 REM STATEMENT VAN BRAM POOT OMGEWERKT TOT FUNKTIE
 30 REM  Marcel Wagenaar
 40 REM  REGIO ROTTERDAM
 50 PRINT $21
 60 P=A;GOSUB a
 70 ?#E8=0
 80 P=A;GOSUB a
 90 PRINT $6
100 IF ?#E8 THEN PRINT "'OUT OF RANGE'";END
110 $T="LINE";T=T+LEN(T)
120 ?T=start/256|#80
130 T?1=start%256;T?2=#80
140 T=T+2;A=P;T!1=A
150 END
160
170a[
180.TA #9400
190:OSWRCH    =#FFF4
200:OSCRLF    =#FFED
210:OSWRNM    =#F7F1
220:LEESEXPR  =#C80B
230:VERHUIS   =#C99F
240:STACK     =#80
250:TP        =#12
260:CR        =#0D
270
280
290:start
300 JSR LEESEXPR
310 JSR initiaal
320 DEY
330
340:zoekregel
350 JSR pasvecaan
360 INY
370 LDA (STACK),Y
```

```
380 BMI brk
390 INY
400 CMP #24,X
410 BNE zoekregel
420 LDA (STACK),Y
430 CMP #15,X
440 BNE zoekregel
450 LDA STACK
460 STA #52
470 LDA STACK+1
480 STA #53
490 DEC #04
500 LDY @#52
510 JMP VERHUIS
520
530:initiaal
540 LDA TP
550 STA STACK+1
560 LDY @#00
570 STY STACK
580 STY #13
590 STY #15
600 STY #14
610 RTS
620
630:pasvecaan
640 INY
650 LDA (STACK),Y
660 CMP @CR
670 BNE pasvecaan
680 CPY #13
690 BCC herschrijfvec1
700 STY #13
710
720:herschrijfvec1
730 INC #14
740 BNE herschrijfvec2
750 INC #15
760
770:herschrijfvec2
780 TYA
790 CLC
800 ADC STACK
810 STA STACK
820 BCC herschrijfvec3
830 INC STACK+1
840
850:herschrijfvec3
860 LDY @#00
870 RTS
880
890:brk
900 LDA @127
910 PHA
920 PHA
930 JMP#ABD5
940]
950 RETURN
```

Een extra label: (.

Wel eens gehoord van fouten in een interpreter? Waarschijnlijk wel, daar dit nog wel eens voorkomt in andere computers, en dit regelmatig in de diverse bladen vermeld staat.
Bekruipt u al een onbehagelijk gevoel, dat deze kopregel ook van toepassing is op onze ACORN ATOM?
U heeft het goed gevoeld, het gaat over een kleine schoonheidsfout in onze basic interpreter. Toch is deze fout niet merkbaar en kan men er zelfs (in bepaalde omstandigheden) goed gebruik van maken.
Voor de acornnisten onder ons die weinig van machine taal weten eerst een kleine uiteenzetting, over vergelijken in deze taal.

Een programma voorbeeld:

```
C52C CMP@#61
C52E BCC #C4F7
C530 SBC@#61
C532 CMP@#1B
C534 BCS #C4F6
C536 INY
C537 ASL
C538 TAX
```

Dit is tevens het stukje waar het over gaat, en waar ook de fout in zit.
Het betreffende stukje is te vinden in ATOM-WARE deel 1 blz 63.
Dit kleine beetje machinecode behandelt het uitzoeken van het getal wat in de accu staat, of het tot de labels hoort of niet.

Nu dezelfde instrukties met commentaar:

```
CMP@#61 vergelijk met 1e waarde label (label a);
BCC      spring weg indien waarde kleiner is,   wanneer niet ge-
        sprongen wordt is de carry geset;
SBC@#61 trek de beginwaarde eraf, zodat bij label a 0 over is,
        bij label b 1 over is, enz.;
CMP@#1B vergelijk met maximaal getal, dit is het aantal labels;
BCS     spring weg indien waarde groter of gelijk is.
```

De rest wordt uitgevoerd als de test geslaagd is.
Waar zit nu de fout? De fout is dat het alfabet 26 letters lang is, of dit nu gewone karakters zijn of geinverteerde.
Het getal #1B geeft 27 karakters, dus is er nog een label dat door de test komt, en wel de geinverteerde haak "(".
Dit feit wordt niet alleen hier uitgevoerd, maar ook in de andere routines.
Bijvoorbeeld de routine die een label moet opzoeken bij de start van een programma, en als het label nog niet geweest (bekend) is.

De fout is dus niet zo ernstig als u wel dacht bij de start van dit verhaal.
Wie heeft er nu ooit geprobeerd om de geinverteerde haak als label te gebruiken? Niemand toch.

We kunnen zelfs van dit 'foutje' profiteren en het extra label gebruiken in programma's.
Daar dit niet zomaar gaat, zal ik nog een kleine uitleg geven.

Wanneer de basic 'runt' en hij komt ons label tegen in een GOTO of een GOSUB, wordt er eerst gekeken of ons label bekend is.
Is het label al gepasseerd, dan is ook het sprong adres bekend. Stel nu dat dit het geval is, dan gaat het zondermeer goed. Is dit niet het geval, en dit komt vaker voor dan andersom, dan wordt het label opgezocht (zie atom-ware blz 86).
Niets aan de hand zult u zeggen, juist, maar zo werkt het niet. Wanneer u de eerste keer 'runt' worden alle label bewaar-adressen op Ø gezet. Daar u wel eens iets in het programma hebt kunnen wijzigen, zouden alle label-adressen niet meer kloppen.

Dit op Ø resetten heeft nog een andere bedoeling, en wel het testen OF er al een adres in staat. Dus of het label al gepasseerd is, weet u nog wel (zie atom-ware blz 85 onderaan).
Dit is de bedoeling van de ORA instruktie, en u voelt het al aankomen: bij ons label is er niets op Ø gezet, en staat er dus 'iets' in. Dus wordt ons label niet opgezocht tijdens de test en wordt tijdens het programma bij aanroep van het label ergens heen gesprongen waar GEEN basic is, met als gevolg een 'hangende atom'.
Dit nu is gemakkelijk te verhelpen door in ons bewaaradres een Ø te poken, en dit zijn de eerste twee adressen na de officiele label adressen (zie atom-ware blz 45 en 46).
Nu ziet u dat dit de adressen #3C1 en #3C2 zijn, maar die worden al gebruikt bij het tekenen, zodat ons label alleen te gebruiken is als er niet geplot wordt.

Tot slot nog een klein 'demootje' om een en ander uit te testen.

```
10 REM LABEL TEST
20 !#3C1=0      bewaaradres label, op nul
30 GOSUB (      spring naar subroutine
40 END          einde programma
50 REM ----------
60( PRINT"TEST OK"'
70 RETURN
```

Dit is uiteraard ook met GOTO van toepassing.

DICK PROTZMAN

*** MEER-KOLOMS DISASSEMBLER ***

Vooral bij lange disassembler-listings is het gewenst om meerdere kolommen naast elkaar af te kunnen drukken.
Dit bevordert de overzichtelijkheid, het spaart papier en de printer kan de listing sneller afdrukken.
Om de kolombreedte te beperken wordt niet de hexadicimale inhoud van de geheugenplaatsen afgedrukt. Dit gebeurt echter wel bij niet bestaande instructies (???).
Ligt in het laatste geval de inhoud tussen #20 en #80, dan wordt tevens de bijbehorende ASCII-code afgedrukt.
Dit kan makkelijk zijn om verborgen data-tabellen te herkennen.
Door op deze manier te werk te gaan blijft de langste instructie beperkt to 16 karakters, zodat op een 80-koloms printer 4 kolommen naast elkaar afgdrukt kunnen worden.
Het een en ander kan verduidelijkt worden met het volgende voorbeeld: een stukje disassembler-listing van het begin van de JOSBOX in 3 kolommen.

```
A000 RTI              A006 DEY              A00C BMI #A024
A001 ??? #BF          A007 INX              A00E CMP (#05),Y
A002 LDX @#FF         A008 INY              A010 BEQ #A007
A004 LDY #5E          A009 LDA #A032,X      A012 DEX
```

Het aantal regels bepaalt de lengte van een bladzijde; na dat aantal volgt een FORM-FEED ($12). Wil men de kantlijn niet naar rechts verschuiven, dan een 0 intoetsen. De minimale kolombreedte bedraagt 17. Als men steeds gebruik maakt van dezelfde lay-out is het misschien handiger om in de regels 420 t/m 450 vaste waarden te programmeren.
Het belangrijkste gedeelte va het programma is de subroutine LL0 die gedeeltelijk overgenomen is uit de JOSBOX. Deze subroutine disassembleert de instructie waarvan het adres op de geheugenplaatsen #92 en #93 staat. Na beeindiging bevat geheugenplaats #91 de lengte van de instructie (1, 2 of 3 byte) en bevat geheugenplaats #90 het aantal karakters dat nodig is om de instructue af te drukken.
Het bepalen van de lengte van de instructie gebeurt door subroutine LL18, die ook zelfstandig aangeroepen kan worden.
Subroutine LL0 maakt gebruik van een stukje data-tabel in de monitor-rom (#F155 -#F1D5) en van een data-tabel in de JOSBOX (#A753-#A7B0).
Het beste kan men deze laatste data-tabel copieeren naar een stukje RAM-geheugen, zodat het programma ook werkt zonder JOSBOX. In mijn programma heb ik de tabel gecopieerd naar #4F00-#4F5D, waarvan het begin aangegeven wordt door variable T in regel 10.
Is de JOSBOX ingeschakeld, dan verplaatst men de tabel als volgt;
COPY #A753,#A7B0,#4F00

Heeft men geen JOSBOX, dan volgt hier de tabel:

```
4F00: 04 20 54 30 0D 80 04 90
4F08: 03 22 54 33 0D 80 04 90
4F10: 04 20 54 33 0D 80 04 90
4F18: 04 20 54 3B 0D 80 04 90
4F20: 00 22 44 33 0D C8 44 00
```

```
4F28: 11 22 44 33 ØD C8 44 A9
4F30: Ø1 22 44 33 ØD 8Ø Ø4 9Ø
4F38: Ø1 22 44 33 ØD 8Ø Ø4 9Ø
4F40: 26 31 87 9A Ø1 22 82 83
4F48: Ø1 Ø1 5A 4E 92 93 87 4B
4F50: 86 9E 2C 29 2C 4Ø 28 23
4F58: 59 ØØ 58 23 23 ØØ
```

Zorg er voor om variable T aan te passen indien men voor een ander geheugengebied kiest. Gebruikt men dit programma altijd met ingeschakelde JOSBOX of AXR1-BOX, dan dus T gelijk maken aan #A753.

Tot slot nog enkele opmerkingen:

* In mijn programma wordt geassembleert vanaf #4CØØ. Dit kan veranderd worden in regel 3Ø door P een andere waarde te geven.

* Het programma kan aanzienlijk ingekort worden als de regels 4Ø-4ØØ geassembleerd zijn. Men heeft dan alleen nog maar de regels 4ØØ (uitgezonderd RTS];) t/m 59Ø nodig. Vul dan voor LLØ en LL18 (regels 51Ø en 55Ø) wel de juiste adressen in.

* Wil men aan het eind de printer uitschakelen, dan voor de instructie END in regel 49Ø toevoegen:P.$12;

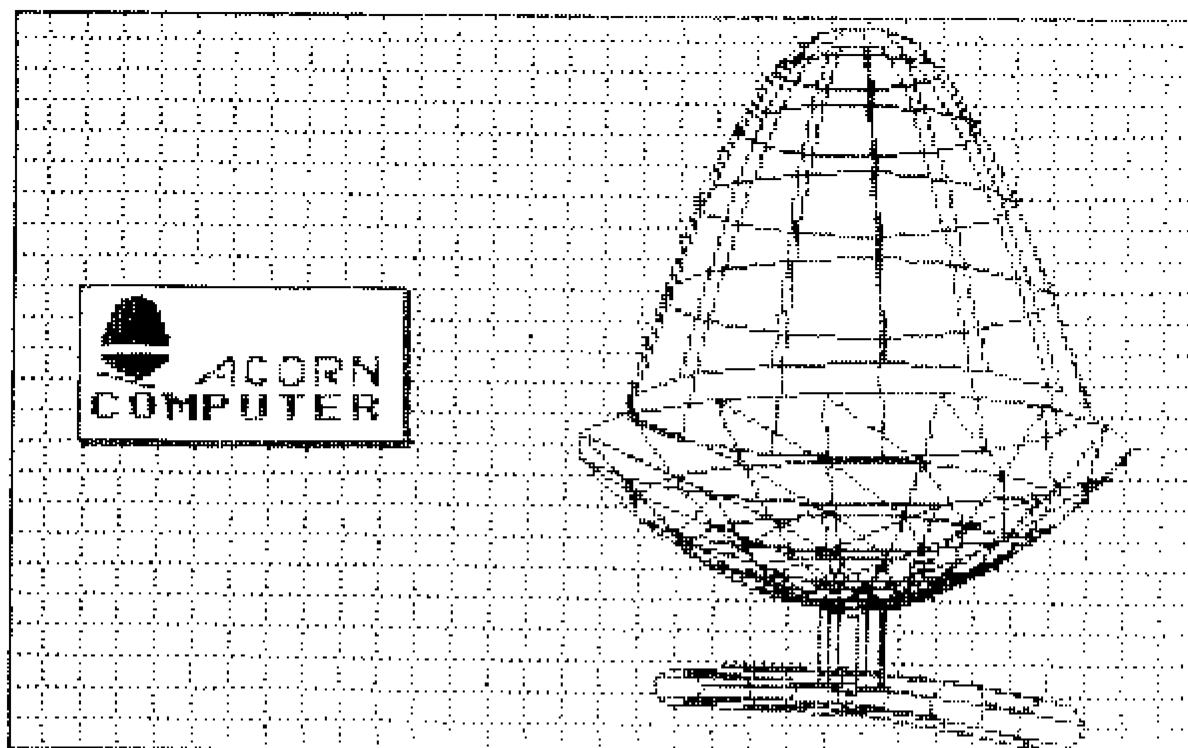
* Wil men de printer inschakelen na het invoeren van de lay-out gegevens, dan P.$2 in regel 4ØØ verwijderen en toevoegen aan regel 48Ø.

```
 10 DIM LL25;T=#4FØØ;P.$3$21
 20 F.Q=ØTO4Ø;LLQ=#4CØØ;N.;F.Q=ØTO1
 30 P=#4CØØ
 40[;:LLØLDA@9;STA#9Ø;JSRLL18;TYA
 50 AND@#8F;TAX;TYA;LDY@3;CPX@#8A
 60 BEQLL3;:LL1LSRA;BCCLL3;LSRA
 70:LL2LSRA;ORA@#2Ø;DEY;BNELL2;INY
 80:LL3DEY;BNELL1;PHA;JSRLL23;PLA
 90 TAY;LDA#F155,Y;STA#69;LDA#F195,Y
100 STA#6A;LDX@3;:LL4LDA@Ø;LDY@5
110:LL5ASL#6A;ROL#69;ROLA;DEY;BNELL5
120 ADC@#3F;JSR#FFF4;DEX;BNELL4
130 JSRLL15;JSR#F7FD;LDX@6;LDY#91;DEY
140:LL6CPX@3;BEQLL1Ø;:LL7ASL#F;BCCLL8
150LDAT+#51,X;JSR#FFF4;JSRLL25
160 LDAT+#57,X;BEQLL8;JSR#FFF4
170 JSRLL25;:LL8DEX;BNELL6;RTS
180:LL9DEY;BMILL7;JSR#F8Ø2;JSRLL24
190:LL1ØLDA#F;CMP@#E8;LDA(#92),Y
200 BCCLL9;JSRLL12;TAX;INX;BNELL11
210:LL11TYA;JSR#F8Ø2;JSRLL24;TXA
220 JSR#F8Ø2;JSRLL24;:LL12LDY#93;TAX
230 BPLLL13;DEY;:LL13ADC#92;BCCLL14
240 INY;:LL14RTS;:LL15CMP@#3F;BEQLL16
```

```
250 RTS;:LL16 JSR#F7FD;LDY@0;LDA@#23
260 JSR#FFF4;LDA(#92),Y;PHA;JSR#F7FA
270 JSRLL24;JSRLL24;PLA;BMILL17
280 CMP@#20;BMILL17;TAX;LDA@#22;PHA
290 JSR#FFF4;TXA;JSR#FFF4;PLA
300 JSR#FFF4;JSRLL24;JSRLL25;:LL17PLA
310 PLA;RTS;:LL18LDX@0;LDA(#92,X);TAY
320 LSRA;BCCLL19;RORA;BCSLL21;CMP@#A2
330 BEQLL21;AND@#87;:LL19LSRA;TAX
340 LDAT,X;BCCLL20;LSRA;LSRA;LSRA
350 LSRA;:LL20 AND@#F;BNELL22
360:LL21 LDY@#80;LDA@0;:LL22TAX
370 LDAT+#44,X;STA#F;AND@3;STA#91;RTS
380:LL23 LDA#93;JSR#F802;LDA#92
390 JMP#F7FA;:LL24INC#90;:LL25INC#90
400 RTS;];N.;P.$6$2
410 P.''"** meerkoloms disassembler**"'''
420 DO IN."AANTAL REGELS"R;U.R>1A.R<133
430 DO IN."AANTAL KOLOMMEN"K;U.K>0A.K<13
440 DO IN."KOLOMBREEDTE"L;U.L>16A.L<81
450 DO IN."KANTLIJN VERSCHUIVEN"V;U.V>-1A.V<21
460 DO IN."BEGINADRES"B;U.B>0A.B<#10000
470 DO IN."EINDADRES"E;U.E>0A.E<65537A.E-B>0
480 DIMRRK
490aRR1=B;IF B>E;END
500 !#92=B;F.S=1TOK
510 F.Q=1TOR;LINKLL18;!#92=!#92+?#91
520 N.;RR((S+1)%(K+1))=!#92;N.
530 P.$12;F.Q=1TOR;IF V>0;GOS.c
540 F.S=1TOK;IF RRS>E;GOS.d;G.b
550 !#92=RRS;LINKLL0;RRS=RRS+?#91
560 F.T=1 TO(L-?#90);P." ";N.
570bN.;P.$13;P.$10;N.;B=RRK;G.a
580cF.T=1TOV;P." ";N.;R.
590dF.T=1TOL;P." ";N.;R.
600 END;REM W.FEIJER BOLIVIASTRAAT 24 DELFT
```

```
 10 PROGRAM ACORN BEELD
 20 REM DATA ACORN
 30B=#9800;E=#9852
 40B!0=#A951;B!2=#A9A9;B!4=#A9A9;B!6=#A9A9;B!8=#BBA9
 50B!10=#BBBB;B!12=#22B0;B!14=#1188;B!16=#1111;B!18=#1111
 60B!20=#B8B8;B!22=#8A8A;B!24=#AAAA;B!26=#999A;B!28=#11B9
 70B!30=#A919;B!32=#B999;B!34=#BBBB;B!36=#B8B8;B!38=#A888
 80B!40=#AAAA;B!42=#1122;B!44=#1111;B!46=#BB11;B!48=#BBBB
 90B!50=#11B1;B!52=#8A11;B!54=#8A8A;B!56=#A999;B!58=#8AAA
100B!60=#8888;B!62=#1111;B!64=#1111;B!66=#1111;B!68=#BB9B
110B!70=#BBBB;B!72=#AA8A;B!74=#8A8A;B!76=#A88A;B!78=#BBBB
120B!80=#B3BB;B!82=#5204
130REM EIKEL
140B=#9852;E=#989A
150B!0=#7000;B!2=#100;B!4=#FC;B!6=#FE03;B!8=#300
160B!10=#FE;B!12=#FF07;B!14=#700;B!16=#FF;B!18=#FF07
170B!20=#F00;B!22=#80FF;B!24=#FF0F;B!26=#F80;B!28=#80FF
180B!30=#FF1F;B!32=#1FC0;B!34=#C0FF;B!36=#FF1F;B!38=#C0
190B!40=#0;B!42=#0;B!44=#1F00;B!46=#C0FF;B!48=#FF1F
200B!50=#FC0;B!52=#80FF;B!54=#FF07;B!56=#300;B!58=#FE
210B!60=#FC38;B!62=#700;B!64=#30;B!66=#E000;B!68=#0
220B!70=#1E;B!72=#AA00
230 GRMOD;?#E1=0;P." "
240 GOS.b
250 FOR A=0 TO 6;SCROLL R;N.
260 BASE #3A
270 CREATE/P:1 0,1,0,1,0,1,0,#AA
280 PAINT 100,100,1
290 PAINT 255,100,1
300 BLOCK 0,14,80,70,40
310 CUBE 1,0,14,80,70,40
320 CUBE 1,3,15,79,70,40
330 VTAB 8;HTAB 2
340 PRINT "COMPUTER"
350 A=#8942
360 B=#9852
370 FOR I=0 TO 23
380   FOR J=0 TO 2
390     A?(I*32+J)=B?(I*3+J)
400   NEXT J
410 NEXT I
420 MOVE 30,94
430 DRAW 33,91
440 MOVE 39,94
450 SHAPE #9800
460 CUBE1,0,0,0,255,191
470 DO U.0
480 REM END
490
```

```
500bREM 3D EIKEL VAN A.de Bruin uit Acorn User
510 DIM AA(100),BB(100),CC(100),DD(100)
520a
530 FOR O=0 TO 3
540  K=0;L=125
550  IFO=0;X=75;S=175;%S=(2*PI/13);R=6;V=4;F=0;H=50
560  IFO=1;X=65;S=185;%S=(2*PI/15);R=5;V=4;F=1;H=60
570  IFO=2;X=120;S=134;%S=1;R=2;V=R;F=0;H=F
580  IFO=3;X=0;S=14;%S=(2*PI)/7;R=5;V=1;F=0;H=0;L=188
590  FOR J=1 TO R
600   B=S-X;GOS.c;X=X+%(H/R);S=S-%(H/R)
610  NEXT J
620 NEXT O
630 R.
640c
650 IFO=0;Y=%(-0.04*((X-125)*(X-125))+190)
660 IFO=1;Y=%((4/360)*((X-125)*(X-125)))+40
670 IFO=2;IFJ=1;Y=40
680 IFO=2;IFJ=2;Y=20
690 IFO=3;Y=%((-10/4115)*((L-125)*(L-125))+19);L=L-20;S=S-1
700 T=((B/2)+L);U=Y;%R=0;Z=0
710o
720 P=%(COS%R*(B/2)+L);Q=%(SIN%R*((B/2)/V)+Y)
730 MOVE T,U;PLOT 5,P,Q;T=P;U=Q
740 IF K%2=0;AAZ=P;BBZ=Q
750 IF K%2<>0;CCZ=P;DDZ=Q
760 IFK=0;G.r
770 IFO=1;GOS.n
780 MOVE AAZ,BBZ;PLOT 5,CCZ,DDZ
790r
800 %R=%R+%S;Z=Z+1
810 FIF %R<(2*PI+.1);G.o
820 AAZ=AA1;BBZ=BB1;CCZ=CC1;DDZ=DD1;K=K+1
830 RETURN
840n
850 IFK%2=0;MOVECCZ,DDZ;PLOT5,AAZ,BBZ;PLOT5,CC(Z+1),DD(Z+1)
860 IFK%2<>0;MOVEAAZ,BBZ;PLOT5,CCZ,DDZ;PLOT5,AA(Z+1),BB(Z+1)
870 R.
```




ACORN
COMPUTER

Een programma voor ATOMISTEN die zelf vliegtuigmotoren willen bouwen.

```
 10 PROGRAM ROTATIAN FAN
 20 P=#2800
 30 DIM VV1;P.$21
 40[
 50 LDA@0;STA#80;LDA@#80
 60 STA#81;LDY@0
 70:VV1 LDA(#80),Y;EOR@#AA
 80 EOR@#FF;STA(#80),Y
 90 INY;BNEVV1;INC#81;LDA#81
100 CMP@#98;BNE VV1;RTS;]
110 GOS.320
120 CLEAR 4
130 C.2;F.Z=0 TO 191
140 MOVE 0,Z;DRAW 128,Z;N.
150 X=740;Y=612;S=450;C=1;N=TRUE
160 Q=1;DO
170 N=NOT N
180 X=X-100;Y=Y-100;S=S-100
190 %A=1.25*PI;DO
200 C=(C+1)&3
210 XIF N THEN T=2-C
220 ELSE T=C
230 Z=T+1;SIDE(X,Y,S,%A,Z)
240 Z=3-T
250 %Z=2.5*PI-%A
260 SIDE(X,Y,S,%Z,Z)
270 %A=%A-PI/20;FU.%A<=PI/4-PI/35
280 Q=Q+1;Z=Q;U.Z=4
290 C=1
300 DO;LI.#2800;F.Z=1TO90;N.
310 UNTIL FALSE
320 PROC SIDE(X,Y,S,%A,C)
330 %V=S*COS%A;%W=S*SIN%A
340 %D=S/3;COLOUR (C%2)
350 %E=%V;%F=%W;%V=%V/6;%W=%W/6
360 E=X;F=Y;X=X/5;Y=Y/5
370 D=%D;%D=%D/4;MOVE(X/2),Y
380 %X=X+%V;%Y=Y+%W;DRAW(%X/2),%Y
390 F.Z=1 TO (%D*2)
400 MOVE(X/2),Y;DRAW(%X/2),%Y
410 %X=%X+0.5;%Y=%Y+0.5
420 N.
430 X=E;Y=F
440 %V=%E;%W=%F
450 %D=D
460 PEND;R.
470f%X=X+%V;%Y=Y+%W
480 F.Z=1 TO %D
490 MOVE(X/2),Y;DRAW(%X/2),%Y
500 %X=%X+1;%Y=%Y+1
510 N.;R.
```

Onderstaand een programma van Bernard Rutgrink om verschillende formaten letters te 'schrijven' in de grafische mode.

```
 10 ?#23=0;?#24=#3B
 20 DIM VV10
 30 F.Z=1TO10;VVZ=-1;N.
 40 F.Z=1TO2
 50 P=#2870;P.$21;[
 60:VV0 LDA#80;STA#8B
 70       CLC;LDA#80;ADC#82
 80       STA#90
 90:VV1 LDA#81;STA#8C
100       CLC;LDA#81
110       ADC#83;STA#91
120:VV2 LDA@0;STA#5B;STA#5D
130       LDA@1;STA#5E
140       LDA#8C;STA#5A
150       LDA#8B;STA#5C
160       LDA#B000;CMP@#F0
170       BNE VV6;JSR #F7AA
180       JMP VV7;:VV6 CMP@#D0
190       BNE VV7;JSR #DFA0
200:VV7 INC#8C;LDA#8C
210       CMP#91;BNE VV2
220       INC#8B;LDA#8B
230       CMP#90;BNE VV1
240       RTS
250:VV3 LDX@0;STX#70
260:VV4 LDA#2860,X
270       STA#90;LDA#84
280       AND#90;BEQ VV5
290       JSR VV0
300:VV5 INC#70;LDX#70
310       SEC;LDA#81;SBC#83
320       STA#81
330       CPX@8;BNE VV4;RTS
340];N.;P.$6
350 !#2860=#08040201
360 !#2864=#80402010;G.470
370bB=#AEE0;DO
380 E=0;F.C=1 TO 7;F=Y
390 D=B?(E+?A)
400 F=F-(E/32*W)
410 ?#81=X;?#80=F;?#82=W
420 ?#83=V;?#84=D;LINK VV3
430 E=E+96;E=E%224;N.
440 A=A+1;X=X+V*9
450 IF?A=32;G.440
460 U.?A=13;R.
470 CLEAR4;ROM0;RESTORE
480 A=#2800;READ $A,X,Y,V,W,C
490 IFX=0;IFY=0;PAU.200;G.710
500 C.C;GOS.b
510 G.480
520 DA."ACORN"
530 DA.16,180,3,2,1
```

```
540 DA."ACORN"
550 DA.17,181,3,2,2
560 DA."COMPUTER"
570 DA.30,130,1,5,2
580 DA."COMPUTER"
590 DA.31,131,1,5,1
600 DA."THIS PROGRAM"
610 DA.15,90,1,2,1
620 DA."IS MADE BY"
630 DA.20,70,1,1,2
640 DA."BERNARD"
650 DA.15,50,2,3,2
660 DA."BERNARD"
670 DA.16,51,2,3,1
680 DA."RUTGRINK"
690 DA.30,20,1,1,2
700 DA."",0,0,0,0,0
710 F.Z=1TO24;SCROLL L;SCROLL U;SCROLL U;N.;G.470
```

Een programma voor AXR1 of JOSBOX bezitters.

```
 10 ON ERR RUN;// POPCORN \\
 20 REM (C) HAMBURGERSOFT
 30 DATA 6,5,6,3,1,3,1,0,6,5,6,3,1,3,1,0
 40 DATA 6,7,8,7,8,6,7,6,7,5,6,5,6,4,6,0
 50 DATA 6,5,6,3,1,3,1,0,6,5,6,3,1,3,1,0
 60 DATA 6,7,8,7,8,6,7,6,7,5,6,5,6,7,8,0
 70 DATA 10,9,10,8,5,8,3,0,10,9,10,8,5,8,3,0
 80 DATA 10,13,12,13,12,10,13,10,11,9,10,9,10,8,10,0
 90 DA.10,9,10,8,5,8,3,0,10,9,10,8,5,8,3,0
100 DATA 10,13,12,13,12,10,13,10,11,9,10,9,10,8,10,0
110 RESTORE
120 DO;READ A
130 GOS.((A*10)+140);U.0
140 PL.R8;R.
150 PL.C8;R.
160 PL.D8;R.
170 PL.E8;R.
180 PL.F8;R.
190 PL.G8;R.
200 PL.A'8;R.
210 PL.B'8;R.
220 PL.C'8;R.
230 PL.D'8;R.
240 PL.E'8;R.
250 PL.F'8;R.
260 PL.G'8;R.
270 PL.F'#8;R.
```

6502 Second Processor voor de ATOM

Multiprocessor - ontwerpen zijn niet nieuw in de computerwereld onder de business- en ontwikkelings-computers. Zelfd de Sinclair QL heeft twee microprocessors, een voor de I/O en de andere voor het rekenwerk. Als voordeel van een twee-microprocessor- ontwerp geldt dat, doordat een gedeelte van de processor-reken- tijd (het I/O gebeuren) wordt overgenomen door de tweede processor, veel sneller kan worden gerekend.
Tevens kan de kloksnelheid worden verhoogd voor de tweede processor. Doordat men bij de tweede processor niet te maken heeft met een specifieke indeling van het geheugen (memory mapped VDU bijvoorbeeld) is het tevens mogelijk om voor andere 6502 systemen geschikte talen/toepassingen te draaien (bijv. BBC basic). Voorwaarde hiervoor is echter dat een communicatie-systeem wordt geschreven dat de twee processors verbindt.
We zijn bij de tweede microprocessor niet gebonden aan de 6502, maar doordat we met onze Atom nu eenmaal de mnemonics van de 6502 kennen is de software eenvoudiger te schrijven.

Het doel van dit schrijven is de belangstelling te polsen voor een 6502 second-processor-kaart voor de Atom.

Een prototype is reeds getest en werkt volledig naar behoren. Als de belangstelling voldoende blijkt te zijn zal er mogelijk een clubkaart van gemaakt worden (dubbelzijdig doorgemetaliseerd). De bedoeling is dat de software in staat zal zijn om zowel BBC als ATOM basic volledig transparant te laten werken (u merkt dus niet dat u in de tweede processor zit).
De kloksnelheid is 2 MHz, zodat de tweede processor-kaart tweemaal zo snel draait als de Atom. Er is 64 kByte ram aan boord waarvan ca. 4 kByte zal worden afgesnoept door het operating systeem, zodat u 60 kByte als werkgeheugen tot uw beschikking hebt. De totale kosten zullen ongeveer fl. 200,00 bedragen, inclusief print, ram, connector en IC-voetjes.
Indien u hiervoor in principe belangstelling heeft kunt u een briefkaart opsturen naar onderstaand adres.
Vermeldt u dan duidelijk erbij of u aan een bouwpakket voldoende heeft of dat u liever een gebouwd en getest geheel zou willen ontvangen.
Als u met meerderen geinteresseerd bent vermeldt u dan tevens het aantal belangstellenden.

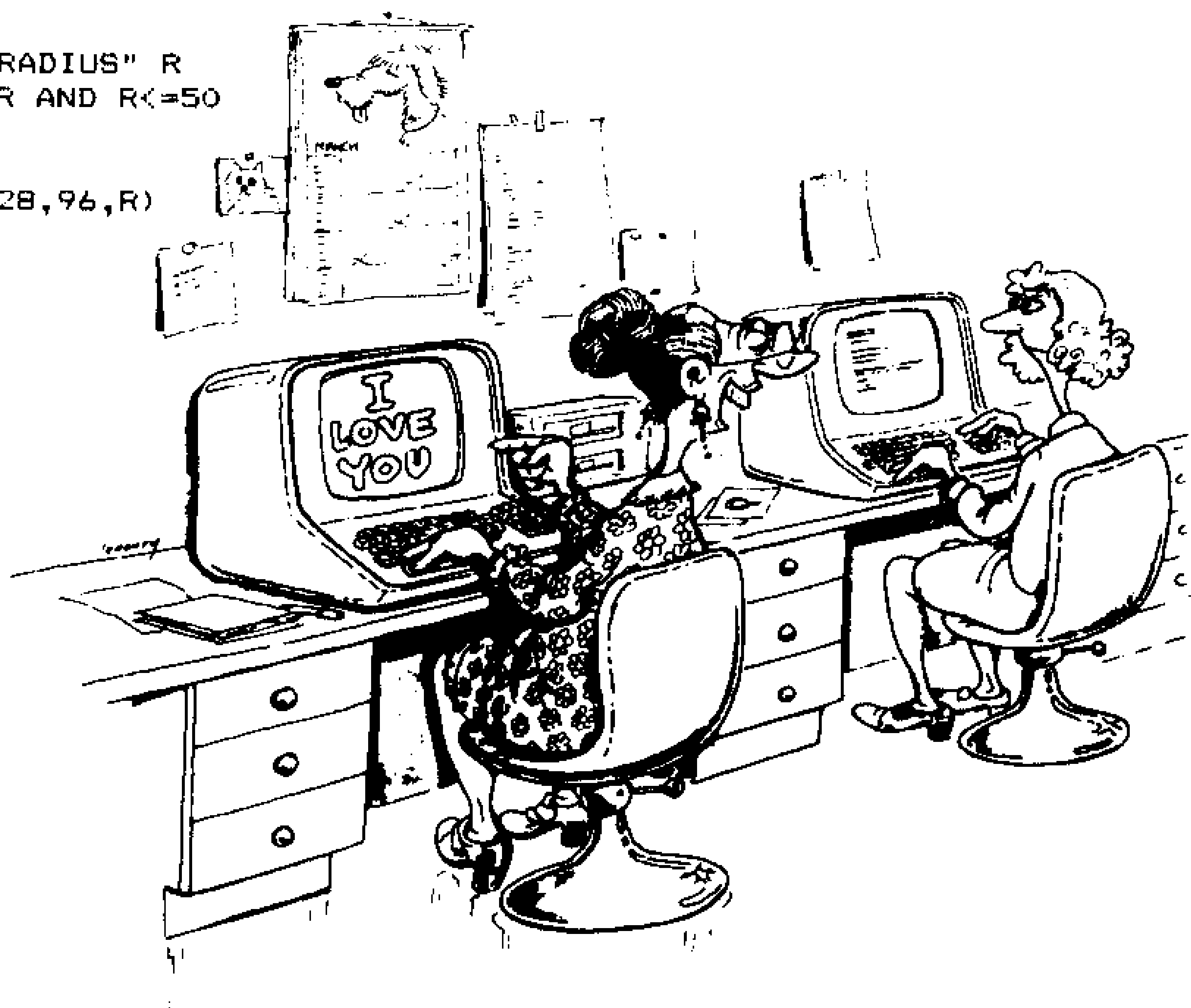
B. Kasteel.

Reakties naar: Redaktie Atom Nieuws
Postbus 1373
8001 BJ Zwolle

Onder vermelding van "2de processor".

Dit korte recursieve programma voor P-CHARME maakt een leuk plaatje.

```
 10 PROGRAM SQUARE
 20
 30 PROC SQUARE(A,B,C),D,E,F,G
 40  XIF C>=1
 50   D=A-C;E=A+C
 60   F=B-C;G=B+C
 70   MOVE D,F
 80   DRAW D,G;DRAW E,G
 90   DRAW E,F;DRAW D,F
100   SQUARE(D,F,C/2)
110   SQUARE(D,G,C/2)
120   SQUARE(E,G,C/2)
130   SQUARE(E,F,C/2)
140  ELSE
150 PEND
160
170 DO
180  INPUT "RADIUS" R
190 UNTIL 0<R AND R<=50
200
210 CLEAR4
220 SQUARE(128,96,R)
230 END
```



Y'see? I still got the ol' sex appeal goin' for me!

Wilt u lid worden van de ATOM COMPUTER CLUB.

Neem dan contact op met de penningmeester van de regio waar u bij ingedeeld wilt worden. Hij kan u vertellen, waar de regiobijeenkomsten zijn en wat u als lid kunt verwachten.

Regio NOORD:
D. Uuldriks Wiemers 14 9642 KG Veendam 05987-19611

Regio OVERIJSSEL/GELDERLAND:
G.J. Noorland Pr. Ireneweg 4 7433 DE Schalkhaar 05700-25294

Regio TWENTE:
W. Verhoeven Witbreuksweg 377-403 7522 ZA Enschede 053 -337026

Regio NOORD-HOLLAND:
P. van Kuik Zuideinde 54-a 1843 JP Groot-Schermer 02997-1902

Regio DEN HAAG:
R. Tiel Vredeoord 96 2544 TZ Den Haag 070 -294170

Regio DELFT:
P. van Alphen H. v. Delftlaan 30 2613 BN Delft 015 -122817

Regio ROTTERDAM:
R. de Haan Brasem 125 2986 HA Ridderkerk 01804-25160

Regio CENTRUM:
P. van Mourik Ruiterstede 60 3431 XN Nieuwegein 03402-48781

Regio ARNHEM:
J. Hartog Keynenbergseweg 60 6871 WK Renkum 08373-13757

Regio ZEELAND:
E. Gijssel Dorpsstraat 86 4424 CZ Wemeldinge 01192-1906

Regio BRABANT-OOST:
P. Ehrlich Roostenlaan 266 5644 BS Eindhoven 040 -114183

Regio LIMBURG:
A. v. Zandvoort Mozartstraat 328 6044 RS Roermond 04750-21797

Regio BELGIE:
R. Leyssens Oude Baan 127 3550 Heusden Belgie

Eventueel kunt u de contributie rechtstreeks overmaken per bank of giro aan de Atom Computer Club te Nuenen. Vermeld echter uw volledige naam, adres en de REGIO waarbij u ingedeeld wilt worden.